

**ON THE INDEPENDENT SPANNING TREE CONJECTURES AND RELATED  
PROBLEMS**

A Dissertation  
Presented to  
The Academic Faculty

By

Alexander Hoyer

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
School of Mathematics

Georgia Institute of Technology

August 2019

Copyright © Alexander Hoyer 2019

# ON THE INDEPENDENT SPANNING TREE CONJECTURES AND RELATED PROBLEMS

Approved by:

Dr. Thomas, Advisor  
School of Mathematics  
*Georgia Institute of Technology*

Dr. Tetali  
School of Mathematics  
*Georgia Institute of Technology*

Dr. Warnke  
School of Mathematics  
*Georgia Institute of Technology*

Dr. Yu  
School of Mathematics  
*Georgia Institute of Technology*

Dr. Vempala  
College of Computing  
*Georgia Institute of Technology*

Date Approved: July 13, 2019

## **ACKNOWLEDGEMENTS**

I would like to thank my advisor, Dr. Robin Thomas, for all of his guidance and dedication towards my academic, professional, and personal development. In addition to his frequent mathematical insights, he has always provided generous portions of his time, attention, and effort towards my research and success, for which I am forever grateful.

I would also like to thank my family and friends for the love and support they have provided me in every stage of my life. I would especially like to thank my wife for her selfless and unwavering support through the challenges of a doctoral program, my mom for her continuous words of encouragement, and my dad both for his advice and for motivating me to earn two degrees.

## TABLE OF CONTENTS

<b>Acknowledgments</b> . . . . .	iii
<b>List of Figures</b> . . . . .	vii
<b>Chapter 1: Introduction and Background</b> . . . . .	1
1.1 Terminology and Notation . . . . .	1
1.2 Conjectures and Theorems . . . . .	3
1.2.1 Independent Spanning Tree Conjectures . . . . .	3
1.2.2 Alternate Approaches . . . . .	5
1.2.3 The Györi- Lovász Theorem . . . . .	5
<b>Chapter 2: The Edge Conjecture for 4-Edge-Connected Graphs</b> . . . . .	6
2.1 Introduction . . . . .	6
2.2 The Chain Decomposition . . . . .	6
2.3 Preliminary Results . . . . .	9
2.4 The Mader Construction . . . . .	12
2.5 Proof of Theorem 2.10 . . . . .	13
2.5.1 Adding an Edge . . . . .	13
2.5.2 Pinching Edges . . . . .	14
2.6 Proof of Theorem 2.11 . . . . .	25

2.7	Summary of Results . . . . .	29
2.8	Algorithms . . . . .	29
2.8.1	Mader Construction . . . . .	30
2.8.2	Chain Decomposition . . . . .	31
2.8.3	Numberings and Trees . . . . .	39
2.8.4	Summary . . . . .	40
<b>Chapter 3: Geometric Approach . . . . .</b>		<b>42</b>
3.1	Introduction . . . . .	42
3.2	Vertex Version . . . . .	46
3.3	Edge Version . . . . .	49
3.4	Conjectures . . . . .	52
<b>Chapter 4: Path-Swapping . . . . .</b>		<b>54</b>
<b>Chapter 5: The Györi-Lovász Theorem . . . . .</b>		<b>61</b>
5.1	Original Theorem . . . . .	61
5.1.1	Terminology and Definitions . . . . .	61
5.1.2	Lemmas . . . . .	62
5.1.3	Proof of Theorem 5.1 . . . . .	65
5.1.4	Algorithm . . . . .	65
5.2	The Generalized Györi-Lovász Theorem . . . . .	67
5.2.1	Terminology and Definitions . . . . .	68
5.2.2	Proof of Theorem 5.6 . . . . .	70

5.3	Fast Computation of Györi-Lovász Partitions . . . . .	74
5.3.1	Introduction . . . . .	74
5.3.2	Proof for $k = 2$ . . . . .	75
5.3.3	Proof for $k = 3$ . . . . .	76
5.3.4	Proof for $k = 4$ . . . . .	79
	<b>References</b> . . . . .	94
	<b>Vita</b> . . . . .	95

## LIST OF FIGURES

2.1	An illustration of an up chain of length 4, a down chain of length 3, and a one-way chain. The red/dashed edges are in earlier chains, while the blue/dotted edges are in later chains. . . . .	7
2.2	An illustration of the procedure in Claim 1. The original up chain $J_1 = J_2$ is on the left, while its replacements in $G'$ are on the right. The red/dashed edges are in earlier chains than $J_1 = J_2$ , while the blue/dotted edges are in later chains than $J_1 = J_2$ . The black/dashed-and-dotted segments represent paths which may have any length (including 0). . . . .	15
2.3	An illustration of the procedure in Claim 4. The original chains $J_1$ and $J_2$ are on the left, while their replacements in $G'$ are on the right. The red/dashed edges are in earlier chains, while the blue/dotted edges are in later chains, with the particular meanings of “earlier” and “later” in the corresponding labels. . . . .	23
2.4	An illustration of the procedure in Claim 5. The original chains $J_1$ and $J_2$ are on the left, while their replacements in $G'$ are on the right. The red/dashed edges are in earlier chains, while the blue/dotted edges are in later chains, with the particular meanings of “earlier” and “later” in the corresponding labels. The black/dashed-and-dotted segments represent paths which may have any length (including 0). . . . .	24
3.1	An illustration of $A(p)$ , $B(p)$ , and $C(p)$ for a point $p$ in the 2-simplex. . . .	43
3.2	The procedure for proving Lemma 3.1 when $Q \subset A(p_0)$ . To choose $p$ , we move vertically from $p_0$ until a member of $Q$ moves out of $A(p)$ . Then we perturb the position of $p$ horizontally so that $p_0$ fills the remaining region with respect to $p$ . . . . .	47
4.1	An illustration of the proof of Claim 7. In $H$ , the bold segment does not intersect any $Q$ paths internally. In $H'$ , edges in the dashed segment which are not used by any $P$ path are deleted. . . . .	56

4.2	An illustration of the proof of Claim 8. In $H$ , the bold segments do not intersect any $Q$ paths internally. In $H'$ , edges in the dashed segments which are not used by any $P$ path are deleted. . . . .	57
4.3	An illustration of the proof of Claim 9. In $H$ , the bold segments do not intersect any $Q$ paths internally. In $H'$ , edges in the dashed segments which are not used by any $P$ path are deleted. . . . .	58
5.1	An example of a configuration. $w_1, w_2, z_1, z_2$ , and $z_3$ are cascade vertices. $R(z_2)$ is shaded. The edge $ab$ is a bridge, and its rank is the rank of $z_3$ . . . .	63
5.2	An example of a configuration. $w_1, w_2, w_3, z_1, z_2$ , and $z_3$ are cascade vertices. $R'(w_2)$ and $R(z_2)$ are shaded. Edges within a subgraph can pass over cascade vertices from left to right, but not from right to left. . . . .	71
5.3	Examples of rank. The edge $ab$ is a bridge. Since $b \in R'(w_3)$ , the rank of the bridge is the rank of $w_3$ . . . . .	72
5.4	Two regions in the graph. If there were no edges from red to green, then the $k - 1$ circled vertices would separate $S$ from $\{v_1, \dots, v_k\}$ , a contradiction. .	73
5.5	(Figure 4 of [7]). An illustration of (a) an up chain, (b) a down chain, (c) an elementary chain, and (d) a triangle chain. . . . .	82



## SUMMARY

We say that trees with common root are (edge-)independent if, for any vertex in their intersection, the paths to the root induced by each tree are internally (edge-)disjoint. The relationship between graph (edge-)connectivity and the existence of (edge-)independent spanning trees is explored. The (Edge-)Independent Spanning Tree Conjecture states that every  $k$ -(edge-)connected graph has  $k$ -(edge-)independent spanning trees with arbitrary root.

We prove the case  $k = 4$  of the Edge-Independent Spanning Tree Conjecture using a graph decomposition similar to an ear decomposition, and give polynomial-time algorithms to construct the decomposition and the trees. We provide alternate geometric proofs for the cases  $k = 3$  of both the Independent Spanning Tree Conjecture and Edge-Independent Spanning Tree Conjecture by embedding the vertices or edges in a 2-simplex, and conjecture higher-dimension generalizations. We provide a partial result towards a generalization of the Independent Spanning Tree Conjecture, in which local connectivity between the root and a vertex set  $S$  implies the existence of trees whose independence properties hold only in  $S$ . Finally, we prove and generalize a theorem of Györi and Lovász on partitioning a  $k$ -connected graph, and give polynomial-time algorithms for the cases  $k = 2, 3, 4$  using the graph decompositions used to prove the corresponding cases of the Independent Spanning Tree Conjecture.

# CHAPTER 1

## INTRODUCTION AND BACKGROUND

The risk posed by bottlenecks is a concern in large and/or unreliable networks. Issues at these points can disconnect the network with relatively few failures. Structural graph theory can provide characterizations of graphs related to connectivity and edge-connectivity, leading to optimal use of networks limited by bottlenecks.

This work concerns such characterizations, particularly related to redundant broadcasting, which can withstand failures. To this end, our primary focus will be the Independent Spanning Tree conjectures. We will also look at the Györi-Lovász Theorem and generalizations, which allow for a sufficiently connected graph to be decomposed into smaller, connected subgraphs.

### 1.1 Terminology and Notation

A *graph*  $G$  is an ordered pair  $(V, E)$ , where  $V = V(G)$  is a finite set whose elements are called *vertices* of  $G$ , and  $E = E(G)$  is a finite multiset whose elements, called *edges* of  $G$ , are multisets of exactly 2 vertices. We will usually denote an edge  $\{u, v\}$  by  $uv$ . Note that the definition of  $E$  allows for an edge containing the same vertex twice (called a *loop*), as well as multiple edges containing the same two vertices (called *parallel edges*). What we are calling a graph is sometimes called a *multigraph*. If  $G$  contains no loops or parallel edges, we call it a *simple graph*.

The elements of an edge  $e$  are called its *endpoints*. We say a vertex  $u$  and an edge  $e$  are *incident* if  $u$  is an endpoint of  $e$ . We say that two vertices  $u$  and  $v$  are *adjacent* if they are both endpoints of a common edge in  $G$ . Similarly, we say that two edges  $e$  and  $f$  are *adjacent* if they have a common endpoint.

The set of all vertices in  $G$  adjacent to a vertex  $u$  is called the *neighborhood* of  $u$ ,

denoted  $\mathcal{N}(u)$ . The *degree* of  $u$ , denoted  $d(u)$ , is the number of edges incident to  $u$ , where loops are counted twice. The minimum and maximum degree among all vertices of  $G$  are denoted  $\delta(G)$  and  $\Delta(G)$ , respectively.

A graph is *k-regular* if every vertex in the graph has degree exactly  $k$ .

A *subgraph* of  $G$  is a graph  $H$  whose vertex set  $V(H)$  and edge set  $E(H)$  are subsets of  $V(G)$  and  $E(G)$ , respectively. An *induced subgraph* is a subgraph  $H$  such that  $E(H)$  contains all edges in  $E(G)$  whose endpoints are both in  $V(H)$ .

If  $G$  is a graph with  $e \in E(G)$  with endpoints  $x, y \in V(G)$ , then the graph  $G/e$  is obtained from  $G$  as follows. Delete  $x$  and  $y$  and create a new vertex  $z$ , then for each deleted edge  $uv$  satisfying  $u \notin \{x, y\}$  and  $v \in \{x, y\}$ , create an edge between  $u$  and  $z$ . This operation is called *contracting* the edge  $e$ .

A graph is *planar* if it can be embedded in  $\mathbb{R}^2$ , mapping vertices to points and edges to curves between those points, so that each edge intersects the rest of the graph exactly at its endpoints. Such an embedding is called a *planar drawing* of the graph.

Let  $G$  be a graph and let  $v_1, v_2, \dots, v_n$  be distinct vertices of  $G$ . We say that the tuple  $(G, v_1, v_2, \dots, v_n)$  is *planar* if there is a planar drawing of  $G$  which lies in a closed disk with  $v_1, v_2, \dots, v_n$  on the boundary of the disc, in this cyclic order.

A *path*  $P$  is a graph with vertices  $V(P) = \{v_1, v_2, \dots, v_n\}$  such that  $E(P) = \{v_1v_2, v_2v_3, \dots, v_{i-1}v_i, \dots, v_{n-1}v_n\}$ . If a path is an induced subgraph, it is an *induced path*. The vertices  $v_1$  and  $v_n$  are called the *ends* of the path, and we say that  $P$  is a path between  $v_1$  and  $v_n$ . A *subpath* of  $P$  is a subgraph of  $P$  which is also a path. We will denote the subpath of  $P$  with ends  $u$  and  $v$  by  $uPv$ .

A *cycle*  $C$  is a graph with vertices  $V(C) = \{v_1, v_2, \dots, v_n\}$  such that  $E(C) = \{v_1v_2, v_2v_3, \dots, v_{i-1}v_i, \dots, v_{n-1}v_n\} \cup \{v_nv_1\}$ . If a cycle is an induced subgraph, it is an *induced cycle*.

A graph  $G$  is *connected* if, for each pair of distinct vertices in  $G$ , there is a path in  $G$  between them. A (*vertex*) *cut*  $A \subset V(G)$  is a set of vertices such that, for some pair of

vertices  $a, b \in V(G) - A$ , every path between  $a$  and  $b$  in  $G$  goes through  $A$ . The *(vertex-)connectivity* of  $G$  is the cardinality of the smallest vertex cut of  $G$ , or  $|V(G)| - 1$  if  $G$  has no vertex cuts. An *edge cut*  $B \subset E(G)$  is a set of edges such that, for some nonempty proper subset  $X \subsetneq V(G)$ ,  $B$  is the set of all edges with one end in  $X$  and the other end in  $V(G) - X$ . The *edge-connectivity* of  $G$  is the cardinality of the smallest edge cut of  $G$ .

A graph  $G$  is *minimally  $k$ -connected* if  $G$  is  $k$ -connected and, for any  $v \in V(G)$ ,  $G - v$  is not  $k$ -connected. Similarly,  $G$  is *minimally  $k$ -edge-connected* if  $G$  is  $k$ -edge-connected and, for any  $e \in E(G)$ ,  $G - e$  is not  $k$ -edge-connected.

A *tree* is a connected graph which does not contain a cycle as a subgraph. A *spanning tree* of  $G$  is a subgraph  $T$  of  $G$  such that  $T$  is a tree and  $V(T) = V(G)$ .

If  $r$  is a vertex of a graph  $G$ , two trees  $T_1, T_2$  of  $G$  are *(vertex-)independent with root  $r$*  if each tree contains  $r$ , and for each  $v \in V(T_1) \cap V(T_2)$ , the unique path in  $T_1$  between  $r$  and  $v$  is internally vertex disjoint from the unique path in  $T_2$  between  $r$  and  $v$ , that is, they do not share any vertices aside from  $r$  and  $v$ . Larger sets of trees are called *(vertex-)independent with root  $r$*  if they are pairwise (vertex-)independent with root  $r$ . Similarly, if the paths in the trees do not share any edges, we say they are *edge-independent with root  $r$* .

## 1.2 Conjectures and Theorems

The following results will be addressed in the following chapters.

### 1.2.1 Independent Spanning Tree Conjectures

Itai and Rodeh [13] conjectured characterizations of the number of vertex- and edge-independent spanning trees in a given graph, and proved the case  $k = 2$  of each.

**Conjecture 1.1.** *If  $G$  is a  $k$ -connected graph and  $r \in V(G)$ , then there exists a set of  $k$  independent spanning trees of  $G$  rooted at  $r$ .*

**Conjecture 1.2.** *If  $G$  is a  $k$ -edge-connected graph and  $r \in V(G)$ , then there exists a set of  $k$  edge-independent spanning trees of  $G$  rooted at  $r$ .*

For convenience, we will refer to Conjectures 1.1 and 1.2 as “the Vertex Conjecture” and “the Edge Conjecture”, respectively.

The conjectures are related to network communication with redundancy. Suppose  $G$  represents a communication network susceptible to node failures. We would like to broadcast information redundantly through multiple trees to combat this susceptibility. Ideally, we would like to choose trees such that a node failure will disconnect each remaining node from at most one copy of the broadcast. The Vertex conjecture states that the absence of node bottlenecks of size less than  $k$  is necessary and sufficient for  $k$  such broadcast copies to be constructable from any source  $r$ . The Edge Conjecture answers the analogous problem where the node failures are not the concern, but rather connections between nodes.

The case  $k = 3$  of the Vertex Conjecture was proven by Cheriyan and Maheshwari [4], and then independently by Zehavi and Itai [20]. Huck [12] proved the case of planar graphs (with any  $k$ ). Building on this work and that of Kawarabayashi, Lee, and Yu [14], the case  $k = 4$  of the Vertex Conjecture was proven by Curran, Lee, and Yu across several papers [6, 5, 7]. The Vertex Conjecture is open for nonplanar graphs with  $k > 4$ .

In 1992, Khuller and Schieber [15] published a later-disproven argument that the Vertex Conjecture implies the Edge Conjecture. Gopalan and Ramasubramanian [10] demonstrated that Khuller and Schieber’s proof fails, but salvaged the technique, and proved the case  $k = 3$  of the Edge Conjecture by reducing it to the case  $k = 3$  of the Vertex Conjecture. Schlipf and Schmidt [19] provided an alternate proof of the case  $k = 3$  of the Edge Conjecture, which does not rely on the Vertex Conjecture. The case  $k = 4$  of the Edge Conjecture is proven in Chapter 2. That is, we will prove the following.

**Theorem 1.3.** *If  $G$  is a 4-edge-connected graph and  $r \in V(G)$ , then there exists a set of four edge-independent spanning trees of  $G$  rooted at  $r$ .*

We also provide an  $O(|E(G)|^2)$  algorithm to construct the edge-independent spanning trees. The case  $k > 4$  of the Edge Conjecture is open.

### 1.2.2 Alternate Approaches

We will introduce two alternate approaches to the Vertex and Edge Conjectures.

In Chapter 3, we consider embedding the vertices (resp. edges) of a graph in a simplex in order to prove the case  $k = 3$  of the Vertex (resp. Edge) Conjecture. We also conjecture that similar embeddings in higher-dimension simplices are possible, which would imply the general Vertex (resp. Edge) Conjecture.

In Chapter 4, we weaken the definition of independence to define trees which have disjoint paths to the root only from a subset of  $V(G)$ . We can then apply double induction, on both  $k$  and the size of the subset where the independence property holds. We are able to prove an analogy of the Vertex Conjecture for general  $k$ , but only for planar graphs and subsets of size 2.

### 1.2.3 The Györi- Lovász Theorem

The following theorem was proven independently Györi [11] and Lovász [16], conjectured and partially solved by Frank [9].

**Theorem 1.4.** *Let  $k \geq 2$  be an integer, let  $G$  be a  $k$ -connected graph on  $n$  vertices, let  $v_1, v_2, \dots, v_k$  be distinct vertices of  $G$ , and let  $n_1, n_2, \dots, n_k$  be positive integers with  $n_1 + n_2 + \dots + n_k = n$ . Then  $G$  has disjoint connected subgraphs  $G_1, G_2, \dots, G_k$  such that, for  $i = 1, 2, \dots, k$ , the graph  $G_i$  has  $n_i$  vertices and  $v_i \in V(G_i)$ .*

We reformulate Györi's proof and generalize the result in Chapter 5. We present a general algorithm for Györi's proof which is  $O^* \left( 2^{2 \left( 1 - \frac{1}{k} \right) n} \right)$ . We also present polynomial-time algorithms for the cases  $k = 2, 3, 4$ , using the same graph decompositions used to prove the corresponding cases of the Vertex Conjecture.

## CHAPTER 2

### THE EDGE CONJECTURE FOR 4-EDGE-CONNECTED GRAPHS

#### 2.1 Introduction

By adapting the technique of Schlipf and Schmidt [19], we prove an edge analog of the planar chain decomposition of Curran, Lee, and Yu [5]. We then use this decomposition to create two edge numberings which define the required trees. Finally, we present a polynomial-time algorithm to find the trees.

#### 2.2 The Chain Decomposition

Throughout this subsection, fix a graph  $G$  with  $|V(G)| \geq 1$  and a vertex  $r \in V(G)$ . We begin by defining a decomposition analogous to the planar chain decomposition in [5].

**Definition 2.1.** An *up chain* of  $G$  with respect to a pair of edge-disjoint subgraphs  $(H, \overline{H})$  is a subgraph of  $G$ , edge-disjoint from  $H$  and  $\overline{H}$ , which is either:

- i A path with at least one edge such that every vertex is either  $r$  or has degree at least two in  $\overline{H}$ , and the ends are either  $r$  or are in  $H$ , OR
- ii A cycle such that every vertex is either  $r$  or has degree at least two in  $\overline{H}$ , and some vertex  $v$  is either  $r$  or has degree at least two in  $H$ . We will consider  $v$  to be both ends of the chain, and all other vertices in the chain to be internal vertices.

Chains which are paths will be called *open* and chains which are cycles will be called *closed*, analogous to the standard ear decomposition.

**Definition 2.2.** A *down chain* of  $G$  with respect to a pair of edge-disjoint subgraphs  $(H, \overline{H})$  is an up chain with respect to  $(\overline{H}, H)$ .

**Definition 2.3.** A *one-way chain* of  $G$  with respect to the pair of edge-disjoint subgraphs  $(H, \overline{H})$  is a subgraph of  $G$ , induced by an edge  $e \notin H \cup \overline{H}$  with ends  $u$  and  $v$ , such that  $u$  is either  $r$  or has degree at least two in  $H$ , and  $v$  is either  $r$  or has degree at least two in  $\overline{H}$ . We call  $u$  the *tail* of the chain and  $v$  the *head*.

**Definition 2.4.** Let  $G_0, G_1, \dots, G_m$  be a sequence of subgraphs of  $G$ . Denote  $H_i = G_0 \cup G_1 \cup \dots \cup G_{i-1}$  and  $\overline{H}_i = G_{i+1} \cup G_{i+2} \cup \dots \cup G_m$ , so that  $H_0$  and  $\overline{H}_m$  are the null graph. We say that the sequence  $G_0, G_1, \dots, G_m$  is a *chain decomposition* of  $G$  rooted at  $r$  if:

1. The sets  $E(G_0), E(G_1), \dots, E(G_m)$  partition  $E(G)$ , AND
2. For  $i = 0, \dots, m$ , the subgraph  $G_i$  is either an up chain, a down chain, or a one-way chain with respect to the subgraphs  $(H_i, \overline{H}_i)$ .

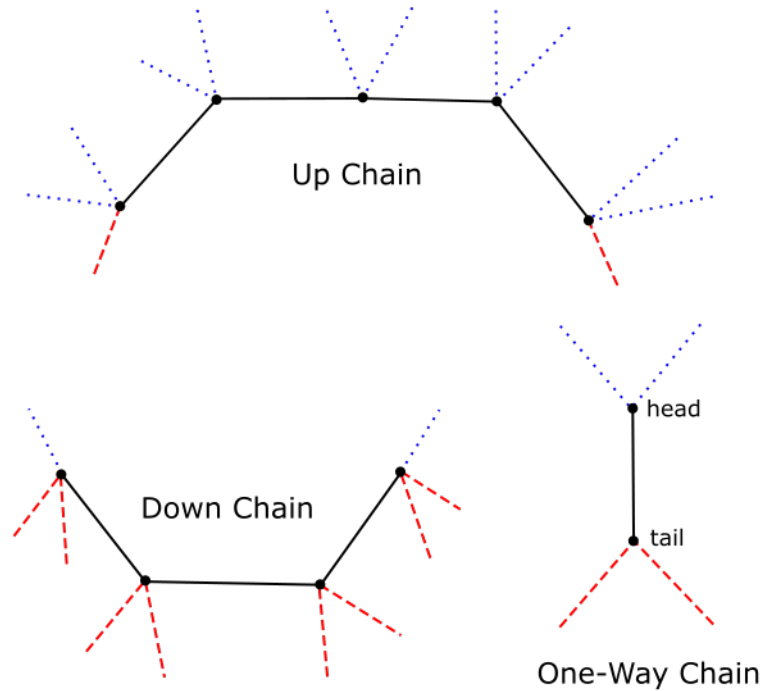


Figure 2.1: An illustration of an up chain of length 4, a down chain of length 3, and a one-way chain. The red/dashed edges are in earlier chains, while the blue/dotted edges are in later chains.



**Definition 2.5.** The *chain index* of  $e \in E(G)$ , denoted  $CI(e)$ , is the index of the chain containing  $e$ .

**Definition 2.6.** An up chain  $G_i$  is **minimal** if no internal vertex of  $G_i$  is in  $\{r\} \cup V(H_i)$ .

**Definition 2.7.** A down chain  $G_i$  is **minimal** if no internal vertex of  $G_i$  is in  $\{r\} \cup V(\overline{H_i})$ .

**Definition 2.8.** A chain decomposition is **minimal** if all of its up chains and down chains are minimal.

**Remarks.**

1. A minimal up chain is analogous to an ear in the standard ear decomposition.
2. The chain decomposition is symmetric in the following sense. If  $G_0, G_1, \dots, G_m$  is a chain decomposition rooted at  $r$ , then  $G_m, G_{m-1}, \dots, G_0$  is a chain decomposition rooted at  $r$ , with the up and down chains switched and the heads and tails of one-way chains switched. Throughout this chapter, we will refer to this fact as “symmetry”.
3.  $G_0$  is either a closed up chain ending at  $r$  or a one-way chain with  $r$  as the tail, and  $G_m$  is either a closed down chain ending at  $r$  or a one-way chain with  $r$  as the head.
4. In the planar chain decomposition in [5], up chains and down chains are analogous to the corresponding open chains. The elementary chain is analogous to a one-way chain.

**Remark 2.9.** An up chain or down chain may be subdivided into several minimal chains by breaking at the offending internal vertices. These minimal chains may then be inserted consecutively to the decomposition at the index of the old chain. In this way, one can easily obtain a minimal chain decomposition from any chain decomposition.

We will prove Theorem 1.3 by combining the following results:

**Theorem 2.10.** *If  $G$  is a 4-edge-connected graph and  $r \in V(G)$ , then  $G$  has a chain decomposition rooted at  $r$ .*

**Theorem 2.11.** *Suppose  $G$  is a graph with no isolated vertices. If  $G$  has a chain decomposition rooted at some  $r \in V(G)$ , then there exists a set of four edge-independent spanning trees of  $G$  rooted at  $r$ .*

### 2.3 Preliminary Results

While not needed for our main results, the following proposition demonstrates how the chain decomposition fits in with the various decompositions used in other cases of the Independent Tree Conjecture and Edge-Independent Tree Conjecture. A partial chain decomposition and its complement are “almost 2-edge-connected” in the following sense.

**Proposition 2.12.** *Suppose  $G_0, G_1, \dots, G_m$  is a chain decomposition of a graph  $G$  rooted at  $r$ . Then for  $i = 1, \dots, m$ ,  $H_i$  and  $\overline{H_{i-1}}$  are connected. Further, if  $e$  is a cut edge of  $H_i$  (resp.  $\overline{H_{i-1}}$ ), then  $e$  induces a one-way chain and one component of  $H_i - e$  (resp.  $\overline{H_{i-1}} - e$ ) contains one vertex and no edges.*

*Proof.* By symmetry, we need only prove the result for the  $H_i$ ’s. The connectivity follows from the fact that every type of chain is connected and contains at least one vertex in an earlier chain.

Suppose  $e$  is a cut edge of some  $H_i$ . Since  $e$  is an edge in  $H_i$ , we have  $CI(e) < i$  and  $H_{CI(e)} \subset H_i$ . We also know that  $H_{CI(e)}$  is connected by the previous paragraph. Then  $e$  cannot be part of an up chain, or else  $e$  would be part of a cycle formed by the chain  $G_{CI(e)}$  and a path in  $H_{CI(e)}$  between the ends of  $G_{CI(e)}$  (if  $G_{CI(e)}$  is open; else the chain itself is a cycle). Also,  $e$  cannot be part of a down chain, or else  $e$  would be part of a cycle formed by  $e$  and a path in  $H_{CI(e)}$  between the ends of  $e$ . Therefore,  $e$  induces a one-way chain.

Let  $C$  be the component of  $H_i - e$  not containing  $r$ , and suppose for the sake of contradiction that  $C$  contains an edge. Let  $e'$  be an edge of  $C$  with minimal chain index. Consider

$G_{CI(e')}$ , the chain containing  $e'$ . Regardless of the chain type, some vertices in  $V(G_{CI(e')})$  are incident to at least two edges in  $H_{CI(e')} \subset H_i$  since  $r \notin C$ , so one of these edges is not  $e$ . This contradicts the minimality of  $CI(e')$ .  $\square$

The next lemma and its corollary will allow us to ignore the possibility of loops in the graph when convenient.

**Lemma 2.13.** *Suppose  $G_0, G_1, \dots, G_m$  is a chain decomposition of  $G$  rooted at  $r$ . If  $v \neq r$  is in  $H_i$  (resp  $\overline{H_i}$ ), then  $v$  is incident to a non-loop edge in  $H_i$  (resp  $\overline{H_i}$ ). If  $v$  has degree at least two in  $H_i$  (resp.  $\overline{H_i}$ ), then  $v$  is incident to two distinct non-loop edges in  $H_i$  (resp.  $\overline{H_i}$ ).*

*Proof.* Note that the second claim in the lemma implies the first, since a loop increases the degree of a vertex by 2, so it suffices to prove the second claim in the lemma.

Suppose  $v$  is incident to a loop, which by symmetry we may assume is in  $H_i$ . Of all loops incident to  $v$ , choose the one with minimal chain index  $j < i$ . Consider the chain classification of  $G_j$ . The chain definitions all coincide for a loop, and require that  $v(\neq r)$  has degree at least two in  $H_j$ . By the minimality of  $j$ ,  $v$  is not incident to any loops in  $H_j$ . It follows that  $v$  is incident to two distinct non-loop edges in  $H_j \subset H_i$ .  $\square$

**Corollary 2.14.** *Suppose  $G_0, G_1, \dots, G_m$  is a chain decomposition of  $G$  rooted at  $r$ , and  $e \in E(G_i)$  is a loop. Then  $G_0, G_1, \dots, G_{i-1}, G_{i+1}, \dots, G_m$  is a chain decomposition of  $G - e$  rooted at  $r$ . Further, if  $G$  has no isolated vertices, then  $G - e$  has no isolated vertices.*

*Proof.* The first claim follows from the preceding lemma. For the second, observe that if  $e$  is the only edge incident to its end, then it fails the conditions for every chain definition.  $\square$

Next, we prove the following useful fact about minimal chain decompositions.

**Lemma 2.15.** *Suppose  $G$  is a graph with no isolated vertices,  $G_0, G_1, \dots, G_m$  is a minimal chain decomposition of  $G$  rooted at  $r$ , and  $v \in V(G)$  with  $v \neq r$ . Then there are indices  $i, j$  so that  $v$  has degree exactly two in  $H_i$  and  $\overline{H_j}$ .*

*Proof.* By symmetry, we need only find  $i$ . Since  $G$  has no isolated vertices,  $v$  is in some chain. Consider the chain  $G_{i_0}$  containing  $v$  so that  $i_0$  is minimal. Note that  $v \notin V(H_{i_0})$ .

If  $G_{i_0}$  is an up chain, then  $v$  is an internal vertex of  $G_{i_0}$  since  $v \notin V(H_{i_0})$ , so  $v$  has degree two in  $G_{i_0}$  and degree at least two in  $\overline{H_{i_0}}$ . Therefore  $\overline{H_{i_0}}$  is not null, so  $i_0 < m$ . Then  $i = i_0 + 1$  completes the proof.

The chain  $G_{i_0}$  is not a down chain since  $v \notin V(H_{i_0})$ .

So we may assume that  $G_{i_0}$  is a one-way chain, and  $v$  must be the head since  $v \notin V(H_{i_0})$ . Therefore  $v$  has degree at least two in  $\overline{H_{i_0}}$ , so we may consider the next chain to contain  $v$ , say  $G_{i_1}$ . Note that  $v$  has degree one in  $H_{i_1}$  by the definition of  $i_1$ .

If  $G_{i_1}$  is an up chain, then it is open and  $v$  is an end of the chain, since the chain decomposition is minimal and  $v$  has degree one in  $H_{i_1}$ . The chain  $G_{i_1}$  is not a down chain since  $v$  has degree one in  $H_{i_1}$ . If  $G_{i_1}$  is a one-way chain, then  $v$  is the head since  $v(\neq r)$  does not have degree at least two in  $H_{i_1}$ . In all cases,  $v$  has degree one in  $G_{i_1}$  and degree at least two in  $\overline{H_{i_1}}$ . Therefore  $\overline{H_{i_1}}$  is not null, so  $i_1 < m$ . Then  $i = i_1 + 1$  completes the proof.  $\square$

Finally, we show that the chain decomposition implies a minimum degree result.

**Lemma 2.16.** *Suppose  $G$  is a graph with no isolated vertices,  $G_0, G_1, \dots, G_m$  is a chain decomposition of  $G$  rooted at  $r$ , and  $v \in V(G)$  with  $v \neq r$ . Then  $v$  has degree at least 4.*

*Proof.* By Corollary 2.14, we may assume that there are no loops in  $G$ . If  $v$  is in an up chain  $G_i$ , then  $v$  has degree at least 2 in  $\overline{H_i}$ , and either degree 2 in  $G_i$  (if  $v$  is internal) or degree at least 1 in  $G_i$  and degree at least 1 in  $H_i$  (if  $v$  is an end). Either way,  $v$  has degree at least 4 in  $G$ . By symmetry, the same is true if  $v$  is in a down chain.

So we may assume that the only chains containing  $v$  are one-way chains. Since  $G$  has no isolated vertices, there is at least one such chain  $G_j$ . Then  $v$  has degree 1 in  $G_j$  and degree at least 2 in  $H_j$  (if  $v$  is the tail) or  $\overline{H_j}$  (if  $v$  is the head). We conclude that  $v$  has degree at least 3 in  $G$ .

Assume for the sake of contradiction that  $v$  does not have degree at least 4. Then  $v$  has degree 3 and is in exactly three one-way chains, say  $G_{\ell_1}, G_{\ell_2}, G_{\ell_3}$  with  $\ell_1 < \ell_2 < \ell_3$ . Consider  $G_{\ell_2}$ . Since we know all of the chains containing  $v$ , we can say that  $v$  has degree 1 in  $H_{\ell_2}$  and degree 1 in  $\overline{H_{\ell_2}}$ . This contradicts the definition of a one-way chain, as  $v$  can be neither the head nor the tail of the chain  $G_{\ell_2}$ . We conclude that  $v$  has degree at least 4 as desired.  $\square$

**Remark.** If  $|V(G)| \geq 2$  in addition to  $G$  having a chain decomposition and no isolated vertices, then  $G$  is 4-edge-connected so  $r$  has degree at least 4 as well. However, we will not need this result, and it will follow from Corollary 2.21.

## 2.4 The Mader Construction

We will adapt the strategy of Schlipf and Schmidt [19] in order to construct a chain decomposition. In particular, we will use a construction method for  $k$ -edge-connected graphs due to Mader [17]. We limit our description of the construction to the needed case  $k = 4$ , since the method is more complicated for odd  $k$ .

**Definition 2.17.** A *Mader operation* is one of the following operations:

1. Add an edge between two (not necessarily distinct) vertices.
2. Consider two distinct edges, say  $e_1$  with ends  $x, y$  and  $e_2$  with ends  $z, w$ , and “pinch” them as follows. Delete the edges  $e_1$  and  $e_2$ , add a new vertex  $v$ , then add the new edges  $e_x, e_y, e_z, e_w$  with one end  $v$  and the other end  $x, y, z, w$  respectively. While  $e_1$  and  $e_2$  must be distinct, the ends  $x, y, z, w$  need not be. In this case,  $v$  will have parallel edges to any repeated vertex.

**Theorem 2.18** ([17, Corollary 14]). *A graph  $G$  is 4-edge-connected if and only if, for any  $r \in V(G)$ , one can construct  $G$  in the following way. Begin with a graph  $G^0$  consisting of  $r$  and one other vertex of  $G$ , connected by four parallel edges. Then, repeatedly perform Mader operations to obtain  $G$ .*

**Remark.** Mader does not explicitly state that one can include a fixed vertex  $r$  in  $G^0$ , but it follows from his work. His proof starts with  $G$ , and then reverses one of the Mader operations while maintaining 4-edge-connectivity. An edge can be deleted unless  $G$  is minimally 4-edge-connected, in which case he finds two vertices of degree 4 in his Lemma 13. He then shows that any degree 4 vertex can be “split off” (the reverse of a pinch) in his Lemma 9, so we can always split off a vertex not equal to  $r$ .

## 2.5 Proof of Theorem 2.10

Due to Theorem 2.18, it suffices to prove that a chain decomposition can be maintained through a Mader operation. The decomposition in the starting graph  $G^0$  is as follows. Two of the edges form a closed up chain. The remaining two edges form a closed down chain.

Suppose the graph  $G'$  is obtained from the graph  $G$  by a Mader operation, with both graphs 4-edge-connected. Assume that we have a chain decomposition  $G_0, G_1, \dots, G_m$  of  $G$ . By Remark 2.9, we may assume that we have a minimal chain decomposition. We wish to create a new chain decomposition of  $G'$ .

### 2.5.1 Adding an Edge

Suppose  $G'$  is obtained from  $G$  by adding an edge with ends  $u, v$ . If one of the ends is the root  $r$ , we can classify the new edge as a one-way chain with tail  $r$  at, say, the very beginning of the chain decomposition. The head must have at least two incident edges in later chains, since all chains are later.

If neither end is  $r$ , choose the minimal index  $i$  such that  $u$  or  $v$  has degree exactly two in  $H_i$ , guaranteed to exist by Lemma 2.15. Note that  $i \geq 1$  since  $H_0$  is null. Without loss of generality,  $u$  has degree exactly two in  $H_i$ . By the definition of  $i$ ,  $v$  has degree at most two in  $H_i$ , and therefore degree at least two in  $\overline{H_{i-1}}$ . We classify the new edge as a one-way chain with tail  $u$  and head  $v$ , between the chains  $G_{i-1}$  and  $G_i$ .

We consider the impact of these changes on other chains in the graph. A new chain was

added, but none of the other chains changed index relative to each other. Vertices may have increased degree in the  $H_i$ 's or the  $\overline{H_i}$ 's due to the new edge, but increasing degree does not invalidate any chain types. Note that some chains may no longer be minimal, so the new chain decomposition in  $G'$  is not necessarily minimal.

### 2.5.2 Pinching Edges

Suppose  $G'$  is obtained from  $G$  by pinching the edges  $e_1$  with ends  $x, y$  and  $e_2$  with ends  $z, w$ , replacing them with edges  $e_x, e_y, e_z, e_w$ . We will use the notation  $J_1 = G_{CI(e_1)} = P_x e_1 P_y$  for the chain containing  $e_1$ , where  $P_x$  is the subpath between  $x$  and an end of  $J_1$  so that  $e_1 \notin E(P_x)$ , and  $P_y$  is defined similarly. Note that  $P_x$  (resp.  $P_y$ ) may have no edges if  $x$  (resp.  $y$ ) is an end of  $J_1$ . In the same way, we will use the notation  $J_2 = G_{CI(e_2)} = P_z e_2 P_w$  for the chain containing  $e_2$ .

We now prove several claims to deal with all possible chain classification and chain index combinations for  $J_1$  and  $J_2$ .

**Claim 1.** *If  $CI(e_1) = CI(e_2)$ , then  $G'$  has a chain decomposition rooted at  $r$ .*

*Proof.* If  $CI(e_1) = CI(e_2)$ , then  $J_1 = J_2$ . Without loss of generality,  $e_1 \in E(P_z)$  and  $e_2 \in E(P_y)$ , so that the chain can be written as  $J_1 = J_2 = P_x e_1 (P_y \cap P_z) e_2 P_w$  (where  $P_y \cap P_z$  may have no edges if  $y = z$ ). Recall that  $e_1$  and  $e_2$  are distinct, so  $J_1 = J_2$  is not a one-way chain.

By symmetry, we may assume  $J_1 = J_2$  is an up chain. In  $G'$ , we replace the chain  $J_1 = J_2$  with the following chains (in the listed order); see Figure 2.2 for an illustration:

1.  $P_x e_x e_w P_w$ . This is an up chain. Since the edges  $e_y$  and  $e_z$  have not yet been used, the new vertex  $v$  is incident to two edges in later chains.
2.  $e_y$ . This is a one-way chain with tail  $v$  and head  $y$ . The tail  $v$  is incident to two edges in earlier chains, namely  $e_x$  and  $e_w$ . The head  $y$  is incident to two edges in later chains since it was an internal vertex in the old up chain  $J_1 = J_2$ .

3.  $e_z$ . This is a one-way chain with tail  $v$  and head  $z$ . The tail  $v$  is incident to two edges in earlier chains, namely  $e_x$  and  $e_w$ . The head  $z$  is incident to two edges in later chains since it was an internal vertex in the old up chain  $J_1 = J_2$ .
4.  $(P_y \cap P_z)$ . Only add this chain if  $P_y \cap P_z$  contains an edge. This is an up chain. The new ends  $y, z$  are each incident to an edge in an earlier chain ( $e_y$  and  $e_z$ , respectively) and are each incident to two edges in later chains since they were interior vertices of the old up chain  $J_1 = J_2$ .

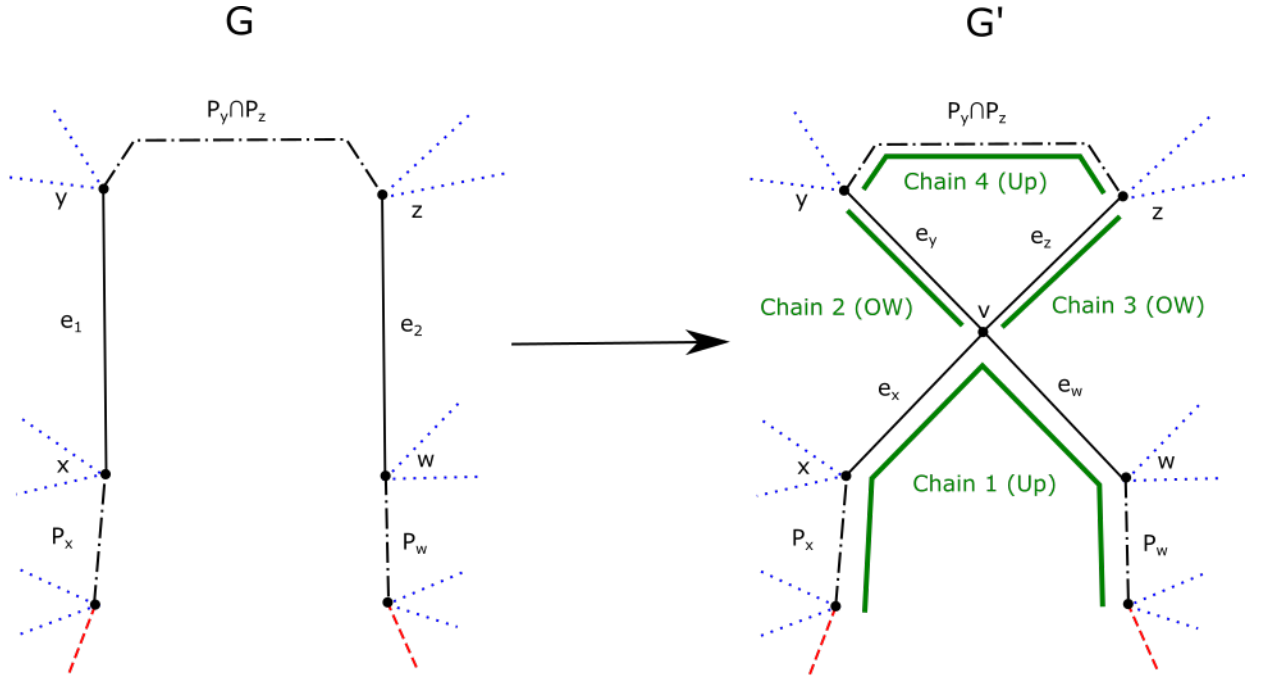


Figure 2.2: An illustration of the procedure in Claim 1. The original up chain  $J_1 = J_2$  is on the left, while its replacements in  $G'$  are on the right. The red/dashed edges are in earlier chains than  $J_1 = J_2$ , while the blue/dotted edges are in later chains than  $J_1 = J_2$ . The black/dashed-and-dotted segments represent paths which may have any length (including 0).

We consider the impact of these replacements on other chains in the graph. We inserted most of the edges of the old chain  $J_1 = J_2$  at the same chain index  $CI(e_1) = CI(e_2)$ , preventing any changes. The exception is the pinched edges  $e_1$  and  $e_2$  which were deleted, but the ends each received new incident edges  $e_x, e_y, e_z, e_w$  inserted at the same chain index



$CI(e_1) = CI(e_2)$ . Thus, we have maintained the chain decomposition. This proves Claim 10.  $\square$

Without loss of generality, we assume the following for the remainder of the proof:

- $CI(e_1) < CI(e_2)$ .
- If  $J_1$  is a one-way chain, then  $x$  is the tail and  $y$  the head.
- If  $J_2$  is a one-way chain, then  $z$  is the tail and  $w$  the head.

**Claim 2.** *Suppose that either  $J_1$  is a one-way chain whose head  $y$  has degree one in  $H_{CI(e_2)}$ , or  $J_2$  is a one-way chain whose tail  $z$  has degree one in  $\overline{H_{CI(e_1)}}$ . Then  $G'$  has a chain decomposition rooted at  $r$ .*

*Proof.* By symmetry, we may assume  $J_1$  is a one-way chain whose head  $y$  has degree one in  $H_{CI(e_2)}$ .

First, we replace  $J_1$  with  $e_x$ . This is a one-way chain with tail  $x$  and head  $v$ . The tail  $x$  was the tail of the old one-way chain  $J_1$ . The head  $v$  has two (in fact three) incident edges in later chains, namely  $e_y, e_z, e_w$ .

- Case 1:  $J_2$  is an up chain. Since  $y$  has degree one in  $H_{CI(e_2)}$ , if  $J_2$  is closed then  $y$  is not the end of  $J_2$ . By swapping  $z$  and  $w$  if necessary, we may assume that  $y$  is not the end of  $J_2$  in  $P_z$ . Thus, the end of  $J_2$  in  $P_z$  is still either  $r$  or incident to an edge in an earlier chain, despite having not placed  $e_y$  yet. We use the edges of  $J_2$  and  $e_y, e_z, e_w$  to construct chains at the index  $CI(e_2)$  as follows:

1.  $P_z e_z$ . This is an up chain. The new end,  $v$ , has one incident edge in an earlier chain ( $e_x$ ) and two incident edges in later chains ( $e_y, e_w$ ). By assumption, the old end in  $P_z$  is still either  $r$  or incident to an edge in an earlier chain.
2.  $e_y$ . This is a one-way chain with tail  $v$  and head  $y$ . The tail  $v$  is incident two edges in earlier chains ( $e_x, e_z$ ). The head  $y$  is either  $r$  or incident to two edges in later chains, since  $y$  has degree one in  $H_{CI(e_2)}$  by assumption.

3.  $e_w$ . This is a one-way chain with tail  $v$  and head  $w$ . The tail  $v$  has two (in fact three) incident edges in earlier chains ( $e_x, e_y, e_z$ ). The head  $w$  is either  $r$  or incident to two edges in later chains, since it was part of the old up chain  $J_2$ .
  4.  $P_w$ . Only add this if  $P_w$  contains an edge. This is an up chain. The new end,  $w$ , has one incident edge in an earlier chain ( $e_w$ ) and two incident edges in later chains since it was an internal vertex of the old up chain  $J_2$ . Since we placed  $e_y$  above, the end of  $J_2$  in  $P_w$  has is either  $r$  or incident to an end in an earlier chain, even if the end is  $y$ .
- Case 2:  $J_2$  is a down chain. Since  $y$  has degree one in  $H_{CI(e_2)}$ ,  $y \notin V(J_2)$ , so each vertex of  $J_2$  is still either  $r$  or incident to two edges in earlier chains, despite having not placed  $e_y$  yet. We use the edges of  $J_2$  and  $e_y, e_z, e_w$  to construct chains at the index  $CI(e_2)$  as follows:
    1.  $P_w$ . Only add this if  $P_w$  contains an edge. This is a down chain. The new end,  $w$ , has one incident edge in a later chain ( $e_w$ ) and two incident edges in earlier chains since it was an internal vertex of the old down chain  $J_1$ .
    2.  $e_w$ . This is a one-way chain with tail  $w$  and head  $v$ . The tail  $w$  is either  $r$  or incident to two edges in earlier chains since it was part of the old down chain  $J_2$ . The head  $v$  is incident to two edges in later chains ( $e_y, e_z$ ).
    3.  $P_z e_z$ . This is a down chain. The new end,  $v$ , has one incident edge in a later chain ( $e_y$ ) and two incident edges in earlier chains ( $e_x, e_w$ ).
    4.  $e_y$ . This is a one-way chain with tail  $v$  and head  $y$ . The tail  $v$  has two (in fact three) incident edges in earlier chains ( $e_x, e_z, e_w$ ). The head  $y$  is either  $r$  or incident to two edges in later chains since  $y$  has degree one in  $H_{CI(e_2)}$  and  $y \notin V(J_2)$  by assumption, so  $y$  has degree at least three in  $\overline{H_{CI(e_2)}}$  unless it is  $r$ .

- Case 3:  $J_2$  is a one-way chain. Since  $y$  has degree one in  $H_{CI(e_2)}$ ,  $y \neq z$  so the tail  $z$  is still either  $r$  or incident to two edges in earlier chains, despite having not placed  $e_y$  yet. We use the edges  $e_y, e_z, e_w$  to construct chains at the index  $CI(e_2)$  as follows:
  1.  $e_z$ . This is a one-way chain with tail  $z$  and head  $v$ . The tail  $z$  is either  $r$  or incident to two edges in earlier chains as discussed above. The head  $v$  is incident to two edges in later chains ( $e_y, e_w$ ).
  2.  $e_w$ . This is a one-way chain with tail  $v$  and head  $w$ . The tail  $v$  is incident to two edges in earlier chains ( $e_x, e_z$ ). The head  $w$  is either  $r$  or incident to two edges in later chains since it was the head of  $J_2$ .
  3.  $e_y$ . This is a one-way chain with tail  $v$  and head  $y$ . The tail  $v$  has two (in fact three) incident edges in earlier chains ( $e_x, e_z, e_w$ ). The head  $y$  is either  $r$  or incident to two edges in later chains since  $y$  has degree one in  $H_{CI(e_2)}$  and  $y \notin V(J_2)$  by assumption, so  $y$  has degree at least three in  $\overline{H_{CI(e_2)}}$  unless it is  $r$ .

We consider the impact of these replacements on other chains in the graph. As before, most of the edges of the old chains  $J_1$  and  $J_2$  were inserted at the same chain indices  $CI(e_1)$  and  $CI(e_2)$  respectively, preventing any changes. The pinched edges  $e_1$  and  $e_2$  were deleted, but the ends  $x, z, w$  each received new incident edges  $e_x, e_z, e_w$  inserted at the same chain indices ( $CI(e_1)$ ,  $CI(e_2)$ , and  $CI(e_2)$  respectively). However,  $e_y$  was inserted at a different chain index than the deleted edge  $e_1$  since  $e_1$  was at  $CI(e_1)$  while  $e_y$  is at  $CI(e_2)$ . By the claim assumptions,  $y$  has degree one in  $H_{CI(e_2)}$ , so there are no chains containing  $y$  between  $CI(e_1)$  and  $CI(e_2)$ , and so no chains were affected by the change. Thus, we have maintained the chain decomposition. This proves Claim 11.  $\square$

We may now assume the following for the remaining cases:

- If  $J_1$  is a one-way chain, then  $y$  has degree at least two in  $H_{CI(e_2)}$ .

- If  $J_2$  is a one-way chain, then  $z$  has degree at least two in  $\overline{H_{CI(e_1)}}$ .

We also make the following conditional definitions, which will aid in distinguishing the remaining cases:

- If  $J_1$  is a one-way chain and  $y$  is not in  $H_{CI(e_1)}$ , then define the minimal index  $i$  such that  $y \in V(G_i)$  and  $CI(e_1) < i < CI(e_2)$ . Since  $i$  is minimal,  $y$  has degree one in  $H_i$  (incident only to the pinched edge  $e_1$ ). From this and the fact that  $G_i$  is a minimal chain, it follows that either  $y$  is one of two distinct ends of the up chain  $G_i$ , or  $y$  is the head of the one-way chain  $G_i$  which is not a loop.
- If  $J_2$  is a one-way chain and  $z$  is not in  $\overline{H_{CI(e_2)}}$ , then define the maximal index  $j$  such that  $z \in V(G_j)$  and  $CI(e_1) < j < CI(e_2)$ . Since  $j$  is maximal,  $z$  has degree one in  $\overline{H_j}$  (incident only to the pinched edge  $e_2$ ). From this and the fact that  $G_j$  is a minimal chain, it follows that either  $z$  is one of two distinct ends of the down chain  $G_j$ , or  $z$  is the tail of the one-way chain  $G_j$  which is not a loop.

**Claim 3.** *Suppose that either one of  $i, j$  is not defined, or  $i < j$ . Then  $G'$  has a chain decomposition rooted at  $r$ .*

*Proof.* The chains replacing  $J_1$  will have indices adjacent to  $CI(e_1)$  and  $i$  (if it is defined). Likewise, the chains replacing  $J_2$  will have indices adjacent to  $CI(e_2)$  and  $j$  (if it is defined). Thus, by the assumptions of this claim, the chains replacing  $J_1$  will have lower chain index than the chains replacing  $J_2$ . This fact will be needed when confirming that the new chains are valid. We begin by replacing  $J_1$  as follows:

- Case 1:  $J_1$  is an up chain. We replace it with  $P_x e_x e_y P_y$ . This is an up chain. The new vertex  $v$  has two incident edges in later chains, namely  $e_z$  and  $e_w$ .
- Case 2:  $J_1$  is a down chain. We replace it with the following chains (in the listed order):

1.  $P_x$ . Only add this chain if  $P_x$  contains an edge. This is a down chain. The new end  $x$  has an incident edge in a later chain, namely  $e_x$ .
  2.  $P_y$ . Only add this chain if  $P_y$  contains an edge. This is a down chain. The new end  $y$  has an incident edge in a later chain, namely  $e_y$ .
  3.  $e_x$ . This is a one-way chain with tail  $x$  and head  $v$ . The tail  $x$  is either  $r$  or incident to two edges in earlier chains since it was in the old down chain  $J_1$ . The head  $v$  has two incident edges in later chains, namely  $e_z$  and  $e_w$ .
  4.  $e_y$ . This is a one-way chain with tail  $y$  and head  $v$ . The tail  $y$  is either  $r$  or incident to two edges in earlier chains since it was in the old down chain  $J_1$ . The head  $v$  has two incident edges in later chains, namely  $e_z$  and  $e_w$ .
- Case 3:  $J_1$  is a one-way chain whose head  $y$  is in  $H_{CI(e_1)}$ . We replace it with the following chains (in the listed order):
    1.  $e_x$ . This is a one-way chain with tail  $x$  and head  $v$ . The tail  $x$  was the tail of the old one-way chain  $J_1$ . The head  $v$  has two (in fact three) incident edges in later chains, namely  $e_y, e_z, e_w$ .
    2.  $e_y$ . This is an up chain. The vertex  $y$  is either  $r$  or incident to two edges in later chains since it was the head of the old one-way chain  $J_1$ , and it has an incident edge in an earlier chain by assumption. The vertex  $v$  has two incident edges in later chains, namely  $e_z$  and  $e_w$ , and is incident to  $e_x$  from the previous chain.
  - Case 4:  $J_1$  is a one-way chain whose head  $y$  is not in  $H_{CI(e_1)}$ . Then  $i$  is defined as above.

First, we replace  $J_1$  with  $e_x$ . This is a one-way chain with tail  $x$  and head  $v$ . The tail  $x$  was the tail of the old one-way chain  $J_1$ . The head  $v$  has two (in fact three) incident edges in later chains, namely  $e_y, e_z, e_w$ .

- Subcase 1:  $y$  is one of two distinct ends of the up chain  $G_i$ . Replace  $G_i$  with  $G_i e_y$ . This is an up chain. Since  $G_i$  was a path and  $v$  is a new vertex, this new chain is a path. The new end  $v$  is adjacent to one edge in an earlier chain ( $e_x$ ) and two edges in later chains ( $e_z$  and  $e_w$ ).
- Subcase 2:  $y$  is the head of the one-way chain  $G_i$  which is not a loop. Then  $y$  is not required to be in  $H_i$  for  $G_i$  to be a valid chain. In fact,  $y$  is not required to be in any of  $H_0, H_1, \dots, H_i$  by the definition of  $i$  and the assumptions of this case. Thus, we can leave  $G_i$  as is and insert the chain  $e_y$  immediately after  $G_i$ . This is an up chain. The vertex  $y$  is incident to an edge in the previous chain  $G_i$ , and is either  $r$  or incident to two edges in later chains since it is the head of  $G_i$ . The vertex  $v$  is adjacent to one edge in an earlier chain ( $e_x$ ) and two edges in later chains ( $e_z$  and  $e_w$ ).

The procedure for replacing  $J_2$  is symmetric, by following the above steps in the reversed chain decomposition.

We consider the impact of these replacements on other chains in the graph. In most cases, we replaced the old chain  $J_1$  with new chains inserted at the same chain index  $CI(e_1)$ , preventing any changes. The pinched edge  $e_1$  was deleted, but the end  $x$  received a new incident edge  $e_x$  at the same chain index  $CI(e_1)$ . In Cases 1-3, the same is true for  $y$ . In Case 4,  $y$  received a new incident edge  $e_y$  either at or immediately after the chain index  $i$ . However, by the definition of  $i$  and the claim assumptions, no chains were affected by the new chain index except  $G_i$ , which was specifically considered and shown to be valid in Case 4. By similar arguments, the changes caused by replacing  $J_2$  also did not invalidate any chains. Thus, we have maintained the chain decomposition. This proves Claim 12.  $\square$

**Claim 4.** *Suppose that both of  $i, j$  are defined and  $i = j$ . Then  $G'$  has a chain decomposition rooted at  $r$ .*

*Proof.* Recall that  $G_i$  is either an up chain or a one-way chain with head  $y$ , and  $G_j$  is either

a down chain or a one-way chain with tail  $z$ . Since  $i = j$ , we conclude that  $G_i = G_j$  must be a one-way chain with tail  $z$  and head  $y$ , and  $y \neq z$  since  $i$  and  $j$  are defined. We can replace  $J_1$  and  $J_2$  with the following chains, in the listed order. The first two will be placed immediately before index  $i = j$ , and the last two immediately after index  $i = j$ ; see Figure 2.3 for an illustration:

1.  $e_x$ . This is a one-way chain with tail  $x$  and head  $v$ . The tail  $x$  was the tail of the old one-way chain  $J_1$  and we are placing this chain after index  $CI(e_1)$ . The head  $v$  has two (in fact three) incident edges in later chains, namely  $e_y, e_z, e_w$ .
2.  $e_z$ . This is a one-way chain with tail  $z$  and head  $v$ . By the definition of  $j$ , the tail  $z$  is either  $r$  or incident to two edges in earlier chains than  $G_j$ , and we are placing this chain immediately before index  $j$ . The head  $v$  has two incident edges in later chains, namely  $e_y$  and  $e_w$ .
3.  $e_y$ . This is a one-way chain with tail  $v$  and head  $y$ . The tail  $v$  has two incident edges in earlier chains, namely  $e_x$  and  $e_z$ . By the definition of  $i$ , the head  $y$  is either  $r$  or incident to two edges in later chains than  $G_i$ , and we are placing this chain immediately after index  $i$ .
4.  $e_w$ . This is a one-way chain with tail  $v$  and head  $w$ . The tail  $v$  has two (in fact three) incident edges in earlier chains, namely  $e_x, e_y, e_z$ . The head  $w$  was the head of the old one-way chain  $J_2$ , and we are placing this chain before  $CI(e_2)$ .

We consider the impact of these replacements on other chains in the graph. The deleted edge  $e_1$  was replaced by two edges with chain index greater than  $CI(e_1)$ , so we must be careful. The edge  $e_x$  was inserted before index  $i$ , but  $x$  had degree at least two in  $H_{CI(e_1)}$ , so losing a degree in later  $H$  subgraphs will not invalidate any chains. The edge  $e_y$  was inserted immediately after index  $i$ , so by the definition of  $i$ , the only chain affected is  $G_i$ . Since  $G_i$  has  $y$  as a head, losing a degree in  $H_i$  will not invalidate the chain. By a symmetric

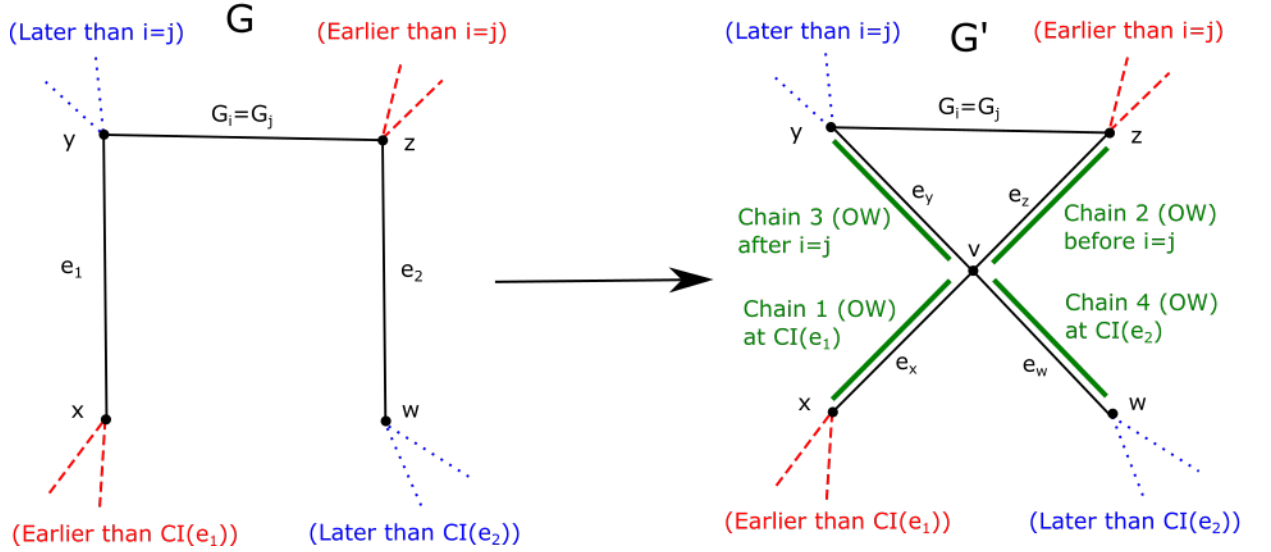


Figure 2.3: An illustration of the procedure in Claim 4. The original chains  $J_1$  and  $J_2$  are on the left, while their replacements in  $G'$  are on the right. The red/dashed edges are in earlier chains, while the blue/dotted edges are in later chains, with the particular meanings of “earlier” and “later” in the corresponding labels.

argument, the changes caused by  $e_z$  and  $e_w$  do not invalidate any chains. This proves Claim 4. □

**Claim 5.** Suppose that both of  $i, j$  are defined, and  $i > j$ . Then  $G'$  has a chain decomposition rooted at  $r$ .

*Proof.* We can replace  $J_1$  and  $J_2$  with the following chains, at the indicated chain indices; see Figure 2.4 for an illustration:

1.  $e_x$ . Add this chain at index  $CI(e_1)$ . This is a one-way chain with tail  $x$  and head  $v$ . The tail  $x$  was the tail of the old one-way chain  $J_1$  and we are placing this chain at index  $CI(e_1)$ . The head  $v$  has two (in fact three) incident edges in later chains, namely  $e_y, e_z, e_w$ .
2.  $e_z$ . Add this chain immediately after  $G_j$ . This is a one-way chain with tail  $z$  and head  $v$ . By the definition of  $j$ , the tail  $z$  is either  $r$  or incident to two edges in earlier



chains than  $G_j$ , and we are placing this chain after index  $j$ . The head  $v$  has two incident edges in later chains, namely  $e_y$  and  $e_w$ .

3.  $e_y$ . Add this chain immediately before  $G_i$ . This is a one-way chain with tail  $v$  and head  $y$ . The tail  $v$  has two incident edges in earlier chains, namely  $e_x$  and  $e_z$ . By the definition of  $i$ , the head  $y$  is either  $r$  or incident to two edges in later chains than  $G_i$ , and we are placing this chain before index  $i$ .
4.  $e_w$ . Add this chain at index  $CI(e_2)$ . This is a one-way chain with tail  $v$  and head  $w$ . The tail  $v$  has two (in fact three) incident edges in earlier chains, namely  $e_x, e_y, e_z$ . The head  $w$  was the head of the old one-way chain  $J_2$ , and we are placing this chain at index  $CI(e_2)$ .

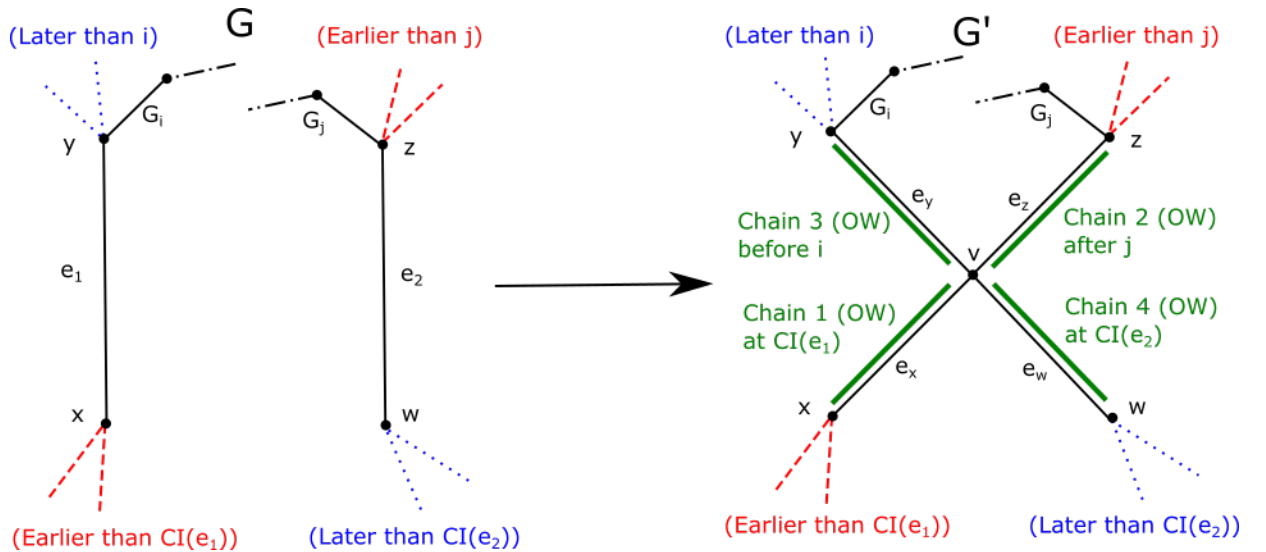


Figure 2.4: An illustration of the procedure in Claim 5. The original chains  $J_1$  and  $J_2$  are on the left, while their replacements in  $G'$  are on the right. The red/dashed edges are in earlier chains, while the blue/dotted edges are in later chains, with the particular meanings of “earlier” and “later” in the corresponding labels. The black/dashed-and-dotted segments represent paths which may have any length (including 0).

We consider the impact of these replacements on other chains in the graph. The edge  $e_1$  was deleted, but  $x$  received a new incident edge  $e_x$  at the same chain index  $CI(e_1)$ . The edge  $e_y$  was inserted before index  $i$ , but the index is still smaller than  $i$ , so by the definition

of  $i$ , no chains are affected. By a symmetric argument, the changes caused by  $e_z$  and  $e_w$  also do not invalidate any chains. This proves Claim 5.  $\square$

The claims cover all possibilities of pinching edges. The proof of Theorem 2.10 is complete. The proof also implies a polynomial-time algorithm to construct a chain decomposition.  $\square$

## 2.6 Proof of Theorem 2.11

Assume that we have a chain decomposition  $G_0, G_1, \dots, G_m$  of  $G$ . By Remark 2.9, we may assume that the chain decomposition is minimal. We will adapt the strategy of Curran, Lee, and Yu [7] to prove Theorem 2.11. In particular, we will construct two partial numberings of the edges of  $G$  using the chain decomposition. We will then construct four spanning trees in two pairs, with one pair associated with each numbering. Within each pair, paths back to the root  $r$  will be monotonic in the associated numbering to ensure independence. Between pairs, paths back to the root  $r$  will be monotonic in chain index to ensure independence.

Using Corollary 2.14, we may assume that there are no loops in  $G$ . By Lemma 2.15, for each vertex  $v \neq r$ , there are two distinct non-loop edges incident to  $v$  whose chain indices are strictly smaller than the chain index of any other edge incident to  $v$ . Likewise there are two distinct edges whose chain indices are strictly larger than the chain index of any other edge adjacent to  $v$ . We will name these edges as follows:

**Definition 2.19.** For each vertex  $v \neq r$ , the two  $f$ -edges of  $v$  are the two incident edges with the lowest chain index. Similarly, the two  $g$ -edges of  $v$  are the two incident edges with the highest chain index.

**Remark 2.20.** By the definition of a down chain, the edges of down chains are never  $f$ -edges. Likewise, by the definition of an up chain, the edges of up chains are never  $g$ -edges.

Next, we will iteratively define a numbering  $f$ , which will assign distinct values in  $\mathbb{R}$  to all edges in up chains and one-way chains. Here, two “consecutive” edges in a chain will

refer to two edges in the chain which are incident to an internal vertex of the chain, so the two edges incident to the end of a closed chain are not consecutive, despite being adjacent.

We begin by numbering the edges in  $E(G_0)$ , and then number the edges of each up chain and one-way chain in order of chain index. When we reach a chain  $G_i$ , we may assume that all edges in  $E(H_i)$  belonging to up chains and one-way chains have been numbered, which includes all  $f$ -edges in  $E(H_i)$  by Remark 2.20. We use the following procedure to number the edges in  $E(G_i)$ :

- If  $G_i$  is a closed up chain containing  $r$ , then number the edges in  $E(G_i)$  so that the values change monotonically between consecutive edges in the chain. The particular numbers used are arbitrary.
- If  $G_i$  is a closed up chain not containing  $r$ , then both  $f$ -edges of the common end have already been numbered. Call these two  $f$ -edges *numbering edges* of  $G_i$ . Say the numbering edges of  $G_i$  have  $f$ -values  $a$  and  $b$ . Number the edges in  $E(G_i)$  so that the values change monotonically between consecutive edges in the chain, and all values are between  $a$  and  $b$ .
- If  $G_i$  is an open up chain containing  $r$ , then  $r$  is an end and the other end is some  $u \neq r$ . At least one  $f$ -edge of  $u$  has already been numbered. Choose an  $f$ -edge which has already been numbered and call it a *numbering edge* of  $G_i$ . Say that  $a$  is the  $f$ -value of the numbering edge. Number the edges in  $E(G_i)$  so that the values increase between consecutive edges in the chain when moving from  $u$  to  $r$ , and all values are larger than  $a$ .
- If  $G_i$  is an open up chain not containing  $r$ , then at least one  $f$ -edge of each end has been numbered. If the ends are  $u$  and  $v$ , we can choose two distinct edges  $e_u, e_v \in E(H_i)$  so that  $e_u$  is an  $f$ -edge of  $u$  and  $e_v$  is an  $f$ -edge of  $v$ . We can choose these two distinct edges because otherwise, the only  $f$ -edge of  $u$  or  $v$  in  $E(H_i)$  would be a single edge between  $u$  and  $v$ , and then  $H_i$  would not be connected. Call the edges

$e_u, e_v$  numbering edges of  $G_i$ . Without loss of generality,  $f(e_u) = a < b = f(e_v)$ . Number the edges in  $E(G_i)$  so that the values increase between consecutive edges in the chain when moving from  $u$  to  $v$ , and all values are between  $a$  and  $b$ .

- If  $G_i$  is a one-way chain whose tail is  $r$ , then number the edge of  $G_i$  arbitrarily.
- If  $G_i$  is a one-way chain whose tail is not  $r$ , then both  $f$ -edges of the tail are already numbered, say with  $f$ -values  $a$  and  $b$ . Number the edge of  $G_i$  between  $a$  and  $b$ .

We symmetrically define a numbering  $g$ , which assigns distinct values in  $\mathbb{R}$  to the edges of down chains and one-way chains, by using the above procedure in the reversed chain decomposition.

We are finally ready to construct the trees. Define the subgraphs  $T_1, T_2, T_3, T_4$  as follows. For each  $v \neq r$ , consider the two  $f$ -edges of  $v$ . Assign the edge with the lower  $f$ -value to  $T_1$  and the edge with the higher  $f$ -value to  $T_2$ . Similarly, consider the two  $g$ -edges of  $v$ . Assign the edge with the lower  $g$ -value to  $T_3$  and the edge with the higher  $g$ -value to  $T_4$ .

Several properties of  $T_1, T_2, T_3, T_4$  will follow from the following claim.

**Claim.** *For any  $v \neq r$ , consider the edge  $e_1$  assigned to  $T_1$  at  $v$ . Let  $v'$  be the other end of  $e_1$ . If  $v' \neq r$ , let  $e'_1$  be the edge assigned to  $T_1$  at  $v'$ . Then  $CI(e'_1) \leq CI(e_1)$  and  $f(e'_1) < f(e_1)$ .*

*Proof.* Let  $e_2$  be the edge assigned to  $T_2$  at  $v$ . The edge  $e_1$  is not in a down chain by Remark 2.20. We break into two cases.

- Suppose  $e_1$  is in an up chain  $G_i$ . Since the chain decomposition is minimal and  $v' \in V(G_i)$ , its  $f$ -edges are either in  $E(G_i)$ , or else have chain index less than  $i$ . In either case,  $CI(e'_1) \leq i = CI(e_1)$  as desired.

Note that  $e_2$  is either in  $E(G_i)$ , or else is the numbering edge of  $G_i$  at the end  $v$ . By the numbering procedure, we know that  $f(e_1)$  is between  $f(e_2)$  and the  $f$ -value of

one of the  $f$ -edges of  $v'$ , say  $e^*$ . By the definition of  $T_1$ ,  $f(e_1) < f(e_2)$ , so it follows that  $f(e^*) < f(e_1)$ . Again by the definition of  $T_1$ ,  $f(e'_1) \leq f(e^*)$ , so  $f(e'_1) < f(e_1)$  as desired.

- Suppose  $e_1$  induces a one-way chain  $G_i$ . Since  $e_1$  is an  $f$ -edge,  $v$  has degree at most one in  $H_i$ , so  $v$  must be the head of  $G_i$ . Then  $v'$  is the tail of  $G_i$ , so the  $f$ -edges of  $v'$  have chain indices smaller than  $i$ , which means  $e'_1 \neq e_1$  and  $CI(e'_1) < CI(e_1)$  as desired.

From the numbering procedure, we know that  $f(e_1)$  is between the  $f$ -values of the two  $f$ -edges of  $v'$ , with  $f(e'_1)$  being the smaller by the definition of  $T_1$ . So,  $f(e'_1) < f(e_1)$  as desired.

In both cases we have  $CI(e'_1) \leq CI(e_1)$  and  $f(e'_1) < f(e_1)$ . This proves the claim.  $\square$

With the claim proven, it follows that the edges assigned to  $T_1$  are all distinct, there are no cycles in  $T_1$ , and following consecutive edges assigned to  $T_1$  produces a path which is decreasing in chain index, strictly decreasing in  $f$ -value, and can only end at  $r$ . Thus,  $T_1$  is connected and is a spanning tree of  $G$ . A similar argument shows that  $T_2$  is a spanning tree of  $G$  where paths to  $r$  are decreasing in chain index and strictly increasing in  $f$ -value. Due to the opposite trends in  $f$ -values,  $T_1$  and  $T_2$  are edge-independent with root  $r$ .

By symmetry, we obtain analogous results for  $T_3$  and  $T_4$ . It remains to show that a tree from  $\{T_1, T_2\}$  and a tree from  $\{T_3, T_4\}$  are edge-independent. The paths back to  $r$  from a vertex  $v \neq r$  are decreasing in chain index in one tree and increasing in chain index in the other tree, but not strictly. The first edges in these paths are an  $f$ -edge and a  $g$ -edge of  $v$ , respectively. By Lemmas 2.15 and 2.16, there is a positive difference in chain index between these initial edges, so the paths are in fact edge-disjoint. The proof of Theorem 2.11 is complete. The proof also implies a polynomial-time algorithm to construct the edge-independent spanning trees.  $\square$

## 2.7 Summary of Results

With Theorems 2.10 and 2.11 proven, we obtain Theorem 1.3. In fact, we can examine the argument more carefully to extract a stronger, summarizing result.

**Corollary 2.21.** *Suppose  $G$  is a graph with no isolated vertices and  $V(G) \geq 2$ . Then the following statements are equivalent.*

1.  $G$  is 4-edge-connected.
2. There exists  $r \in V(G)$  so that  $G$  has a chain decomposition rooted at  $r$ .
3. For all  $r \in V(G)$ ,  $G$  has a chain decomposition rooted at  $r$ .
4. There exists  $r \in V(G)$  so that  $G$  has four edge-independent spanning trees rooted at  $r$ .
5. For all  $r \in V(G)$ ,  $G$  has four edge-independent spanning trees rooted at  $r$ .

*Proof.* Theorem 2.10 gives us  $(1) \Rightarrow (3)$ . Theorem 2.11 gives us  $(2) \Rightarrow (4)$  and  $(3) \Rightarrow (5)$ . Trivially, we have  $(3) \Rightarrow (2)$  and  $(5) \Rightarrow (4)$ . Therefore, we need only show  $(4) \Rightarrow (1)$ .

Assume for the sake of contradiction that  $G$  has four edge-independent spanning trees rooted at some  $r \in V(G)$ , but is not 4-edge-connected. Suppose  $S \subseteq E(G)$  is an edge cut with  $|S| < 4$ . Consider a vertex  $v$  in a component of  $G - S$  not containing  $r$ . Using the paths in each of the edge-independent spanning trees, we find that there exist four edge-disjoint paths between  $v$  and  $r$ . This contradicts the existence of  $S$ .  $\square$

## 2.8 Algorithms

The proof above implies a polynomial-time algorithm to construct four edge-independent spanning trees from a 4-edge-connected graph. We can separate this algorithm into three phases, outlined below.

### 2.8.1 Mader Construction

Recall the two *Mader operations* used in the process (adding edges and pinching edges), discussed above.

**Definition 2.22.** A *Mader construction* of a graph  $G$ , rooted at  $r$ , is a sequence of graphs  $G^0, G^1, \dots, G^p$  such that:

1.  $G^0$  consists of  $r$ , one other vertex, and four parallel edges between them,
2.  $G^p = G$ , and
3. for  $i = 2, 3, \dots, p$ ,  $G^i$  can be obtained from  $G^{i-1}$  by performing a Mader operation

Mader [17] proved that a graph is 4-edge-connected if and only if it has a Mader construction (hence every graph in the sequence is 4-edge-connected as well) and outlined an algorithm to create one. However, the runtime was not analyzed. We start that process with the following lemma.

**Lemma 2.23.** Suppose  $G$  is a minimally 4-edge-connected graph, and  $G'$  is a 4-edge-connected graph obtained from  $G$  by splitting a degree 4 vertex  $v$  and creating edges  $e_1, e_2$ . Then one of  $G', G' - e_1, G' - e_2, G' - e_1 - e_2$  is minimally 4-edge-connected.

*Proof.* Suppose not. Then there is an edge  $e \notin \{e_1, e_2\}$  which is removable in  $G'$  but not in  $G$ . This means there is a 4-edge-cut  $X$  in  $G$  containing  $e$ , but  $e$  is not in any 4-edge-cut of  $G'$ . Also note that since  $e$  is in  $G'$ ,  $v$  is not an endpoint of  $e$ .

Label the sides of the cut  $X$  as  $A$  and  $B$ , and say  $v \in A$  without loss of generality. We must have  $d(v; B) \leq 2$  or else we could create a smaller cut by moving  $v$  to  $B$ , contradicting 4-edge-connectivity.

Since  $v$  is not an endpoint of  $e$ ,  $|A| \geq 2$ , so  $A$  and  $B$  correspond to nonempty sets  $A'$  and  $B'$  in  $G'$ , separated by edge cut  $X'$ . Observe that  $X' - X \subseteq \{e_1, e_2\}$ , and suppose  $|X' - X| = 1$ . Then at least one neighbor of  $v$  is in  $B$ , and consequently  $X - X'$  contains

an edge. Similarly, if  $|X' - X| = 2$ , then  $v$  must have two neighbors in each of  $A$  and  $B$  (counted with multiplicity), and consequently  $|X - X'| = 2$ . In all cases, we have  $|X'| \leq |X|$ , contradicting the fact that  $e$  is removable in  $G'$ .  $\square$

Now we are ready to analyze the first algorithm.

**Theorem 2.24.** *There exists an algorithm with the following specifications:*

**Input:** A 4-edge-connected graph  $G$  on  $n$  vertices and  $m$  edges, with  $r \in V(G)$

**Output:** A Mader construction  $G^0, G^1, \dots, G^p$  of  $G$  rooted at  $r$ .

**Running time:**  $O(m^2)$

*Proof.* The algorithm begins by looping through the  $m$  edges of  $G$  and testing for removability. This can be done by deleting the edge, say  $uv$ , and testing whether the flow between  $u$  and  $v$  in  $G - uv$  is at least 4, which takes  $O(m)$  time. Thus, this entire stage of the algorithm takes  $O(m^2)$  time.

We now have a minimally 4-edge-connected graph. By [17] there are at least two vertices of degree 4, so we can find a degree 4 vertex  $v \neq r$  in  $O(n)$  time. Suppose  $v$  has (not necessarily distinct) neighbors  $x, y, z, w$ .

There are three ways to split  $v$ , and by [17] at least one of them will produce a 4-edge-connected graph. Attempt each one, testing the flow between previous neighbors of  $v$  to confirm edge connectivity. Once a valid split is found, check both of the new edges for removability. By Lemma 2.23, the graph is now minimally 4-edge-connected and we can find another vertex to split. Each split takes  $O(m)$  time to process, and we need to split  $O(n)$  vertices, so this stage of the algorithm takes  $O(mn)$  time.

Once the graph contains two vertices and four edges between them, the algorithm terminates.  $\square$

### 2.8.2 Chain Decomposition

Next, we use the Mader construction to create a chain decomposition of  $G$ .



**Theorem 2.25.** *There exists an algorithm with the following specifications:*

**Input:** A 4-edge-connected graph  $G$  on  $n$  vertices and  $m$  edges, and a Mader construction  $G^0, G^1, \dots, G^p$  of  $G$  rooted at  $r$ .

**Output:** A chain decomposition of  $G$  rooted at  $r$ .

**Running time:**  $O(m^2)$

*Proof.* We will allow the chain indices to be any rational number, so that we may reorder chains more conveniently. We can map these back to integer indices at the end of the algorithm in  $O(m)$  time.

We will store data in the following way. Keep a list of chains ordered by chain index, and for each chain, store its edges and vertices in order of adjacency in the chain. For each vertex, list its incident edges in order of chain index, breaking ties by listing edges from the same chain in the order they are listed in the chain data. For each edge in the chain data, store a pointer to its occurrence in the vertex data for each endpoint.

The algorithm will loop through the  $p = O(m)$  stages of the Mader construction, updating the chain decomposition in each iteration. For the first iteration, construct a chain decomposition of  $G^0$  with exactly two chains. Use (any) two of the edges to define an up chain at index 1, and use the remaining two to define a down chain with index 2. With a fixed number of edges and chains, this step takes  $O(1)$  time.

Before addressing each consecutive Mader operation, minimize the chain decomposition as follows. Loop through each chain, skipping any one-way chains. If some  $G_i$  is an up chain, check each internal vertex for incident edges in  $H_i$ . If such edges are found at vertices  $u_1, u_2, \dots, u_t$ , break the up chain into  $t + 1$  up chains, with consecutive chains sharing an endpoint at each  $u_i$ . For down chains, perform the symmetric equivalent of the same steps (check for incident edges in  $\overline{H}_i$  and breaking into down chains).

Since our chain indices are rational numbers, we can pick an interval around the previous chain index which does not contain any other chain indices, and assign the new chains to indices from the interval. Within this interval, we use the order of the  $G_i$  data to order

the new chains, so that edges do not need to be reordered in the data. We will still need to update the chain indices. With our pointers, we can immediately update the vertex data while updating the chain data, so we use time proportional to the chain length. As the sum of the chain lengths is  $m$ , this phase of the algorithm is  $O(m)$ .

Next, break into cases based on the current Mader operation.

1. If the operation is to add a new edge, label the new edge  $e$  and its ends  $u, v$ . If  $v = r$ , swap the labels of  $u$  and  $v$ .
  - If  $u = r$ , add a new one-way chain before all existing chain indices, with edge  $e$ , tail  $u$ , and head  $v$ . Append the new chain to the beginning of the chain data. Append  $e$  to the beginning of the adjacency lists for  $u$  and  $v$ . This takes  $O(1)$  time. Move on to the next Mader operation.
  - Consider the index  $i_u$  (resp.  $i_v$ ) of the second edge in the adjacency list for  $u$  (resp.  $v$ ). If  $i_u > i_v$ , swap the labels of  $u$  and  $v$ . Add a new one-way chain immediately after index  $i_u$  (before any subsequent indices), with edge  $e$ , tail  $u$ , and head  $v$ . Insert  $e$  into the adjacency lists for  $u$  and  $v$ . Insert the new chain into the chain data. We will take  $O(m)$  time to search the edge/chain lists for correct location to insert the new data. Move on to the next Mader operation.
2. If the operation is to pinch edges, label the pinched edges  $e_1, e_2$ , and the ends  $x, y$  for  $e_1$  and  $z, w$  for  $e_2$ . Label the chains of the pinched edges  $J_1$  for  $e_1$  and  $J_2$  for  $e_2$ . Decompose the chains as  $J_1 = P_x e_1 P_y$  and  $J_2 = P_z e_2 P_w$ , where the path  $P_x$  has one end  $x$  and similarly for  $P_y, P_z$ , and  $P_w$ . Label the new edge incident to  $x$  as  $e_x$ , and similarly  $e_y, e_z$ , and  $e_w$ .
3. If  $J_1 = J_2$ , we may assume by symmetry that  $J_1 = J_2 = P_x e_1 (P_y \cap P_z) e_2 P_w$ .
  - Remove  $J_1 = J_2$ . Replace it with the following new chains. Form an interval around  $CI(e_1) = CI(e_2)$  containing no other indices, and assign the new chains

to indices from the interval, in the listed order:

- Up chain  $P_x e_x e_w P_w$ .
- One-way chain  $e_y$ , with tail  $v$  and head  $y$ .
- One-way chain  $e_z$ , with tail  $v$  and head  $z$ .
- Up chain  $P_y \cap P_z$  (if the intersection does not contain an edge, skip this chain).

- Add the new vertex  $v$  to the vertex data. Update the chain data by removing  $J_1 = J_2$  and inserting four new entries in its place. With our pointers, we can immediately update the vertex data while updating the chain data. We will take  $O(m)$  time to search the lists for correct locations and update the chain indices.
- Move on to the next Mader operation.

4. If  $CI(e_1) > CI(e_2)$ , swap the labels of  $e_1$  and  $e_2$ .

5. If  $J_1$  is one-way with tail  $y$ , swap the labels of  $x$  and  $y$ .

6. If  $J_2$  is one-way with tail  $w$ , swap the labels of  $z$  and  $w$ .

7. Check if at least one of the following conditions is true:  $J_1$  is one-way and  $d(y; H_{CI(e_2)}) = 1$ , or the symmetric equivalent,  $J_2$  is one-way and  $d(z; \overline{H_{CI(e_1)}}) = 1$ . We will take  $O(m)$  time to search the lists for correct locations. If so,

- If first condition above is false, follow the steps below in the reversed chain decomposition.
- Remove  $J_1$ .
- Add a new one-way chain  $e_x$ , with tail  $x$  and head  $v$ , at  $CI(e_1)$ .
- If  $J_2$  is an up chain,
  - If  $y \in P_z$ , swap the labels of  $z$  and  $w$ .

- Remove  $J_2$ . Replace it with the following new chains. Form an interval around  $CI(e_2)$  containing no other indices, and assign the new chains to indices from the interval, in the listed order:
  - \* Up chain  $P_z e_z$
  - \* One-way chain  $e_y$ , with tail  $v$  and head  $y$ .
  - \* One-way chain  $e_w$ , with tail  $v$  and head  $w$ .
  - \* Up chain  $P_w$  (if  $P_w$  does not contain an edge, skip this chain).
- If  $J_2$  is a down chain,
  - Remove  $J_2$ . Replace it with the following new chains. Form an interval around  $CI(e_2)$  containing no other indices, and assign the new chains to indices from the interval, in the listed order:
    - \* Down chain  $P_w$  (if  $P_w$  does not contain an edge, skip this chain).
    - \* One-way chain  $e_w$ , with tail  $w$  and head  $v$ .
    - \* Down chain  $P_z e_z$ .
    - \* One-way chain  $e_y$ , with tail  $v$  and head  $y$ .
- If  $J_2$  is a one-way chain,
  - Remove  $J_2$ . Replace it with the following new chains. Form an interval around  $CI(e_2)$  containing no other indices, and assign the new chains to indices from the interval, in the listed order:
    - \* One-way chain  $e_z$ , with tail  $z$  and head  $v$ .
    - \* One-way chain  $e_w$ , with tail  $v$  and head  $w$ .
    - \* One-way chain  $e_y$ , with tail  $v$  and head  $y$ .
- Add the new vertex  $v$  to the vertex data. Update the chain data by removing  $J_1$  and  $J_2$  and inserting the new entries in its place. With our pointers, we can immediately update the vertex data while updating the chain data, except for

vertex  $y$  since  $e_y$  has moved to  $CI(e_2)$ . We find a new location for  $e_y$  in  $O(m)$  time. We will take  $O(m)$  time to search the lists for correct locations and update the chain indices.

- Move on to the next Mader operation.

8. Try to define  $i, j$ :

- If  $J_1$  is a one-way chain, consider the first two entries in the edge list for  $y$ . If the first entry is  $e_1$  and the second entry is not  $e_2$ , then define  $i$  to be the chain index of the second entry. It takes  $O(m)$  time to find the chain index of the second entry.
- If  $J_2$  is a one-way chain, consider the last two entries in the edge list for  $z$ . If the last entry is  $e_2$  and the second-to-last entry is not  $e_1$ , then define  $j$  to be the chain index of the second-to-last entry. It takes  $O(m)$  time to find the chain index of the second-to-last entry.

9. If at least one of  $i, j$  is undefined or  $i < j$ ,

- If  $J_1$  is an up chain,
  - Remove  $J_1$ . Replace it with up chain  $P_x e_x e_y P_y$ .
- If  $J_1$  is a down chain,
  - Remove  $J_1$ . Form an interval around  $CI(e_1)$  containing no other indices, and assign the new chains to indices from the interval, in the listed order:
    - \* Down chain  $P_x$  (if  $P_x$  does not contain an edge, skip this chain).
    - \* Down chain  $P_y$  (if  $P_y$  does not contain an edge, skip this chain).
    - \* One-way chain  $e_x$ , with tail  $x$  and head  $v$ .
    - \* One-way chain  $e_y$ , with tail  $y$  and head  $v$ .
- If  $J_1$  is a one-way chain and  $y \in H_{CI(e_1)}$ ,

- Remove  $J_1$ . Form an interval around  $CI(e_1)$  containing no other indices, and assign the new chains to indices from the interval, in the listed order:
  - \* One-way chain  $e_x$ , with tail  $x$  and head  $v$ .
  - \* Up chain  $e_y$
- If  $J_1$  is a one-way chain and  $y \notin H_{CI(e_1)}$ ,
  - \* Remove  $J_1$ . Replace it with one-way chain  $e_x$ , with tail  $x$  and head  $v$ .
  - \* If  $G_i$  is an up chain with distinct ends, and  $y$  is one of those ends, replace  $G_i$  with up chain  $G_i e_y$ .
  - \* If  $G_i$  is a non-loop one-way chain with head  $y$ , then add up chain  $e_y$  at an index immediately following  $i$  (before any subsequent indices).
- Replace  $J_2$  by following the above steps in the reversed chain decomposition.
- Add the new vertex  $v$  to the vertex data. Update the chain data by removing  $J_1$  and  $J_2$  and inserting the new entries in its place. With our pointers, we can immediately update the vertex data while updating the chain data. We will take  $O(m)$  time to search the lists for correct locations and update the chain indices.
- Move on to the next Mader operation.

10. If  $i = j$ ,

- Remove  $J_1$  and  $J_2$ . Replace with the following chains:
  - Place the following chains immediately before index  $i$  (after any preceding chains), in the listed order:
    - \* One-way chain  $e_x$ , with tail  $x$  and head  $v$ .
    - \* One-way chain  $e_z$ , with tail  $z$  and head  $v$ .
  - Place the following chains immediately after index  $i$  (before any subsequent chains), in the listed order:
    - \* One-way chain  $e_y$ , with tail  $v$  and head  $y$ .

\* One-way chain  $e_w$ , with tail  $v$  and head  $w$ .

- Add the new vertex  $v$  to the vertex data. Update the chain data by removing  $J_1$  and  $J_2$  and inserting the new entries in its place. We will take  $O(m)$  time to search the lists for correct locations and update the chain indices. We also need to move  $e_x, e_y, e_z, e_w$  to the new indices in the vertex data for  $x, y, z, w$ , respectively, taking  $O(m)$  time to find new locations.
- Move on to the next Mader operation.

11. Else (meaning  $i > j$ ),

- Remove  $J_1$  and  $J_2$ . Replace with the following chains at the specified indices:
  - One-way chain  $e_x$ , with tail  $x$  and head  $v$ , at index  $CI(e_1)$ .
  - One-way chain  $e_z$ , with tail  $z$  and head  $v$ , immediately after index  $j$  (before any subsequent indices).
  - One-way chain  $e_y$ , with tail  $v$  and head  $y$ , immediately before index  $i$  (after any preceding indices).
  - One-way chain  $e_w$ , with tail  $v$  and head  $w$ , at index  $CI(e_2)$ .
- Add the new vertex  $v$  to the vertex data. Update the chain data by removing  $J_1$  and  $J_2$  and inserting the new entries in its place. We will take  $O(m)$  time to search the lists for correct locations and update the chain indices. We also need to move  $e_x, e_y, e_z, e_w$  to the new indices in the vertex data for  $x, y, z, w$ , respectively, taking  $O(m)$  time to find new locations.
- Move on to the next Mader operation.

An iteration of the algorithm takes  $O(m)$  time, and we need to run  $p = O(m)$  iterations to reach  $G$ . So, the algorithm runs in  $O(m^2)$  time.

□

### 2.8.3 Numberings and Trees

Finally, we can build the independent spanning trees from the chain decomposition.

**Theorem 2.26.** *There exists an algorithm with the following specifications:*

**Input:** *A 4-connected graph  $G$  on  $n$  vertices and  $m$  edges, and a chain decomposition of  $G$  rooted at  $r$ .*

**Output:** *An  $f$ -numbering and a  $g$ -numbering of the edges of  $G$  rooted at  $r$ .*

**Running time:**  $O(m)$

*Proof.* We will describe the process for creating the  $f$ -numbering. We can then create the  $g$ -numbering by following the same process in the reversed chain decomposition.

The algorithm loops through the chains of  $G$  in order of chain index, numbering the edges of each  $G_i$  depending on the chain type.

- If  $G_i$  is a closed up chain containing  $r$ , then number the edges in  $E(G_i)$  so that the values change monotonically between consecutive edges in the chain. The particular numbers used are arbitrary.
- If  $G_i$  is a closed up chain not containing  $r$ , identify the  $f$ -edges of the common end (the two incident edges with lowest chain index). These have already been numbered, say with values  $a$  and  $b$ . Number the edges in  $E(G_i)$  so that the values change monotonically between consecutive edges in the chain, and all values are between  $a$  and  $b$ .
- If  $G_i$  is an open up chain containing  $r$ , then it has another end  $u \neq r$ . Choose an  $f$ -edge of  $u$  which is not in  $G_i$ , picking arbitrarily if there are two. This edge has already been numbered, say with value  $a$ . Number the edges in  $E(G_i)$  so that the values increase between consecutive edges in the chain when moving from  $u$  to  $r$ , and all values are larger than  $a$ .



- If  $G_i$  is an open up chain not containing  $r$ , identify the  $f$ -edges of each end, and choose a pair of distinct edges  $e_u, e_v$  so that  $e_u$  is an  $f$ -edge of  $u$  and  $e_v$  is an  $f$ -edge of  $v$ . Without loss of generality, they have values  $a$  and  $b$  respectively with  $a < b$ . Number the edges in  $E(G_i)$  so that the values increase moving from  $u$  to  $v$ , and all values are between  $a$  and  $b$ .
- If  $G_i$  is a down chain, do not number its edges.
- If  $G_i$  is a one-way chain whose tail is  $r$ , then number the edge of  $G_i$  arbitrarily.
- If  $G_i$  is a one-way chain whose tail is not  $r$ , then identify the  $f$ -edges of the tail. These are already numbered, say with values  $a$  and  $b$ . Number the edge of  $G_i$  between  $a$  and  $b$ .

□

**Theorem 2.27.** *There exists an algorithm with the following specifications:*

**Input:** *A 4-connected graph  $G$  on  $n$  vertices and  $m$  edges, and an  $f$ -numbering and a  $g$ -numbering of the edges of  $G$  rooted at  $r$*

**Output:** *Four edge-independent spanning trees of  $G$  rooted at  $r$*

**Running time:**  $O(n)$

*Proof.* For each vertex  $v \neq r$ , consider the  $f$ -edges of  $v$ . Assign the edge with the lower  $f$ -value to  $T_1$ , and the other to  $T_2$ . Similarly, consider the  $g$ -edges of  $v$ . Assign the one with the lower  $g$ -value to  $T_2$ , and the other to  $T_4$ .

Once we have looped through all vertices of  $G$ , the edges assigned to  $T_1, T_2, T_3, T_4$  will induce the four edge-independent spanning trees. □

#### 2.8.4 Summary

With the algorithms above, we can find edge-independent spanning trees in a 4-edge-connected graph as follows.

**Theorem 2.28.** *There exists an algorithm with the following specifications:*

**Input:** *A 4-connected graph  $G$  on  $n$  vertices and  $m$  edges, containing vertex  $r$*

**Output:** *Four edge-independent spanning trees of  $G$  rooted at  $r$*

**Running time:**  $O(m^2)$

*Proof.* Apply Theorem 2.24 to create a Mader construction of  $G$  in  $O(m^2)$  time. Apply Theorem 2.25 to create a chain decomposition of  $G$  in  $O(m^2)$  time. Apply Theorem 2.26 to create edge numberings of  $G$  in  $O(m)$  time. Finally, apply Theorem 2.27 to construct edge-independent spanning trees in  $O(n)$  time.  $\square$

## CHAPTER 3

### GEOMETRIC APPROACH

#### 3.1 Introduction

Our objective is to give an alternate proof technique for the case  $k = 3$  of the Vertex and Edge Conjectures. Our motivation is the ordering of vertices (resp. edges) to solve the case  $k = 2$  of the Vertex (resp. Edge) Conjecture in [13]. We can think of these orderings geometrically as an embedding of objects in a line segment, and extend this concept by embedding in a triangle. We will refer to the geometric triangle as a “2-simplex” to avoid confusion with the graph theory notion of a “triangle”.

We will use barycentric coordinates to parameterize the 2-simplex as  $S = \{(a, b, c) : 0 \leq a, b, c \leq 1 \text{ and } a + b + c = 1\}$ . The corners are  $(1, 0, 0)$ ,  $(0, 1, 0)$ , and  $(0, 0, 1)$ , and the other points are convex combinations of the corners.

Next, for a point  $p = (a_0, b_0, c_0) \in S$  we define:

$$A(p) = \{(a, b, c) \in S : a - a_0 > b - b_0, c - c_0\}$$

$$B(p) = \{(a, b, c) \in S : b - b_0 > a - a_0, c - c_0\}$$

$$C(p) = \{(a, b, c) \in S : c - c_0 > a - a_0, b - b_0\}$$

Note that  $A(p)$ ,  $B(p)$ , and  $C(p)$  are disjoint and span most of  $S$ , missing only the altitudes from  $p$  to each boundary of the 2-simplex. Since we will be considering a finite number of points in general position, we may assume that the three regions span every embedded object. These regions can be visualized in Figure 3.1.

We can define a set of distances describe the regions more conveniently. For  $p_1 = (a_1, b_1, c_1)$  and  $p_2 = (a_2, b_2, c_2)$  define  $d_{AB}(p_1, p_2) = (a_1 - a_2) - (b_1 - b_2)$  and similarly

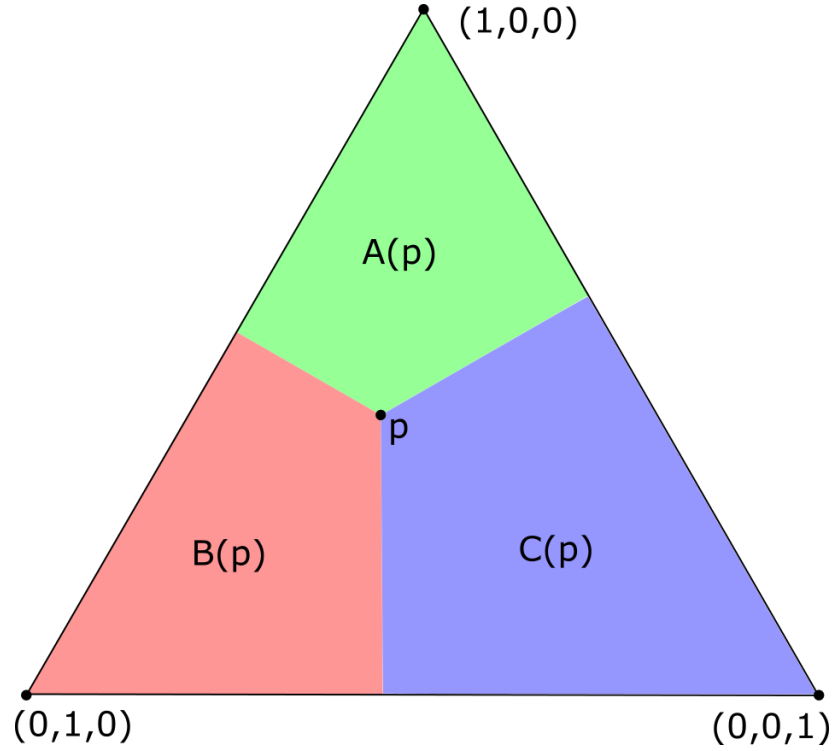


Figure 3.1: An illustration of  $A(p)$ ,  $B(p)$ , and  $C(p)$  for a point  $p$  in the 2-simplex.

for other pairs of coordinates. Then we can characterize the regions as:

$$A(p) = \{q : d_{AB}(q, p) > 0 \text{ and } d_{AC}(q, p) > 0\}$$

$$B(p) = \{q : d_{AB}(q, p) < 0 \text{ and } d_{BC}(q, p) > 0\}$$

$$C(p) = \{q : d_{AC}(q, p) < 0 \text{ and } d_{BC}(q, p) < 0\}$$

Notice that the distances are invariant if we translate all coordinates uniformly, and scale proportionally if we scale all coordinates uniformly. We can use this to effectively ignore the requirement that barycentric coordinates are nonnegative. If negative coordinates are ever generated, we can simply translate coordinates uniformly across the 2-simplex until they are nonnegative, then scale down so that they add to 1 again. Through these transformations, the distances between pairs of points will maintain sign and ordering, meaning that region assignments will be unchanged as well.

The following lemma will be our main tool for embedding objects.

**Lemma 3.1.** *Suppose we have a point  $p_0$  and a finite set of points  $Q \subset S$  with  $p_0 \notin Q$  and  $|Q| \geq 2$ , all in general position. Then for every  $\hat{q} \in Q$  there is a point  $p \in S - Q - \{p_0\}$  in general position satisfying the following conditions.*

1. *At least one member of  $\{p_0\} \cup Q$  is in each of  $A(p), B(p), C(p)$ .*
2. *For each  $q \in Q$ ,  $p$  occupies the same region as  $p_0$  with respect to  $q$ .*
3. *The point  $p$  occupies the same region as  $\hat{q}$  with respect to  $p_0$ .*

*Proof.* Let  $p_0 = (a_0, b_0, c_0)$ . First, suppose that members of  $Q$  occupy at least two different regions with respect to  $p_0$ , say  $\hat{q} \in A(p_0)$  and  $q \in B(p_0)$  without loss of generality. Then we choose  $p = (a_0 + 2\varepsilon, b_0 + \varepsilon, c_0 - 3\varepsilon)$  for some  $\varepsilon > 0$ .

First notice that these are valid coordinates since they sum to  $a_0 + b_0 + c_0 = 1$ . For sufficiently small  $\varepsilon$ , we will have  $\hat{q} \in A(p)$  and  $q \in B(p)$  (since  $\hat{q} \in A(p_0)$  and  $q \in B(p_0)$ ), and  $p$  will be in the same region as  $p_0$  with respect to members of  $Q$ . Further, we can compute

$$d_{AB}(p_0, p) = -\varepsilon = -d_{AB}(p, p_0)$$

$$d_{BC}(p_0, p) = -4\varepsilon = -d_{BC}(p, p_0)$$

$$d_{AC}(p_0, p) = -5\varepsilon = -d_{AC}(p, p_0)$$

This confirms  $p_0 \in C(p)$  and  $p \in A(p_0)$ , as desired.

So, we may assume that all members of  $Q$  are in the same region with respect to  $p_0$ , say  $A(p_0)$ . This tells us that, for any  $q \in Q$ ,  $d_{AB}(q, p_0)$  and  $d_{AC}(q, p_0)$  are both positive. It also gives us symmetry between the  $b$  and  $c$  coordinates, so we may assume without loss of generality that the smallest of these distances is  $d_{AB}(q^*, p_0)$  for some  $q^* = (a^*, b^*, c^*) \in Q$ . For convenience, denote this distance  $3\delta := d_{AB}(q^*, p_0)$ .

Choose  $p = (a_0 + 2\delta + 2\varepsilon_1, b_0 - \delta - \varepsilon_1 + \varepsilon_2, c_0 - \delta - \varepsilon_1 - \varepsilon_2)$  for some  $\varepsilon_1, \varepsilon_2 > 0$ . First notice that these are valid coordinates since they sum to  $a_0 + b_0 + c_0 = 1$ . We now show that, with the right choice of  $\varepsilon_1$  and  $\varepsilon_2$ , we have satisfied the lemma.

We compute the relevant distances between  $p_0$  and  $p$ :

$$\begin{aligned}
d_{AB}(p_0, p) &= [a_0 - (a_0 + 2\delta + 2\varepsilon_1)] - [b_0 - (b_0 - \delta - \varepsilon_1 + \varepsilon_2)] \\
&= -3\delta - 3\varepsilon_1 + \varepsilon_2 \\
d_{AC}(p_0, p) &= [a_0 - (a_0 + 2\delta + 2\varepsilon_1)] - [c_0 - (c_0 - \delta - \varepsilon_1 - \varepsilon_2)] \\
&= -3\delta - 3\varepsilon_1 - \varepsilon_2 \\
d_{BC}(p_0, p) &= [b_0 - (b_0 - \delta - \varepsilon_1 + \varepsilon_2)] - [c_0 - (c_0 - \delta - \varepsilon_1 - \varepsilon_2)] \\
&= -2\varepsilon_2
\end{aligned}$$

For any  $q = (a, b, c) \in Q$ , we can compute:

$$\begin{aligned}
d_{AB}(q, p) &= [a - (a_0 + 2\delta + 2\varepsilon_1)] - [b - (b_0 - \delta - \varepsilon_1 + \varepsilon_2)] \\
&= [(a - a_0) - (b - b_0) - 3\delta] - 3\varepsilon_1 + \varepsilon_2 \\
&= [d_{AB}(q, p_0) - 3\delta] - 3\varepsilon_1 + \varepsilon_2 \\
d_{AC}(q, p) &= [a - (a_0 + 2\delta + 2\varepsilon_1)] - [c - (c_0 - \delta - \varepsilon_1 - \varepsilon_2)] \\
&= [(a - a_0) - (c - c_0) - 3\delta] - 3\varepsilon_1 + \varepsilon_2 \\
&= [d_{AC}(q, p_0) - 3\delta] - 3\varepsilon_1 - \varepsilon_2 \\
d_{BC}(q, p) &= [b - (b_0 - \delta - \varepsilon_1 + \varepsilon_2)] - [c - (c_0 - \delta - \varepsilon_1 - \varepsilon_2)] \\
&= [(b - b_0) - (c - c_0)] - 2\varepsilon_2 \\
&= d_{BC}(q, p_0) - 2\varepsilon_2
\end{aligned}$$

For  $q = q^*$ , the  $AB$  distance simplifies to  $d_{AB}(q^*, p) = -3\varepsilon_1 + \varepsilon_2$ .

First, observe that with  $\varepsilon_1$  and  $\varepsilon_2$  sufficiently small, we will have:

$$d_{AB}(p_0, p) < 0$$

$$d_{AC}(p_0, p) < 0$$

$$d_{BC}(p_0, p) < 0$$

$$d_{AB}(q, p) > 0 \text{ for all } q \in Q - \{q^*\}$$

$$d_{AC}(q, p) > 0 \text{ for all } q \in Q$$

$$\text{sgn}(d_{BC}(q, p)) = \text{sgn}(d_{BC}(q, p_0)) \text{ for all } q \in Q$$

The first three lines tell us that  $p \in A(p_0)$  and  $p_0 \in C(p)$ . The remaining lines tell us that for  $q \neq q^*$ ,  $q \in A(p)$  and  $p$  is in the same region as  $p_0$  with respect to  $q$ . Since we assumed  $Q \subset A(p_0)$  and we already know  $p \in A(p_0)$ , we have satisfied condition 3 without needing to consider  $\hat{q}$  in particular.

We do, however, need to consider  $q^*$ . Choosing  $\varepsilon_2 < 3\varepsilon_1$  gives us  $d_{AB}(q^*, p) < 0$ . Also note that  $d_{BC}(q, p) = d_{AC}(q, p) - d_{AB}(q, p) > 0$ , so  $q^* \in B(p)$ ,  $p \in C(q^*)$ , and  $p_0 \in C(q^*)$ . This satisfies conditions 1 and 2, so the proof is complete.  $\square$

With the lemma proven, we can move on to the main results.

### 3.2 Vertex Version

We will prove the following:

**Theorem 3.2.** *Let  $G$  be a 3-connected graph with  $r \in V(G)$  and  $s_1, s_2 \in \mathcal{N}(r)$ . Then there exists an embedding of  $V(G)$  in a 2-simplex with barycentric coordinates so that  $r = (1, 0, 0)$ ,  $s_1 = (0, 1, 0)$ ,  $s_2 = (0, 0, 1)$ , and for every  $v \in V(G) - \{r, s_1, s_2\}$ ,  $v$  has a neighbor in each of  $A(v)$ ,  $B(v)$ , and  $C(v)$ . Further,  $s_1$  has neighbors in  $A(s_1) - \{r\}$  and  $C(s_1)$ , and  $s_2$  has neighbors in  $A(s_2) - \{r\}$  and  $B(s_2)$ .*

*Proof.* Since multiple edges and loops do not affect vertex connectivity, we may assume  $G$

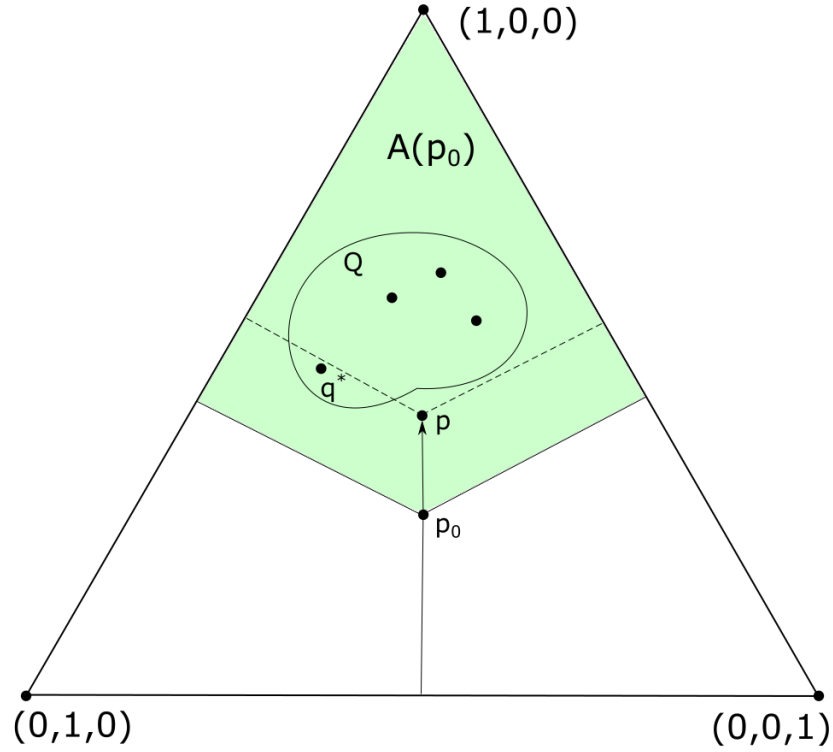


Figure 3.2: The procedure for proving Lemma 3.1 when  $Q \subset A(p_0)$ . To choose  $p$ , we move vertically from  $p_0$  until a member of  $Q$  moves out of  $A(p)$ . Then we perturb the position of  $p$  horizontally so that  $p_0$  fills the remaining region with respect to  $p$ .

is a simple graph. We induct on  $|V(G)|$ . We break into cases based on  $|V(G)|$ , in each case either embedding  $V(G)$  as needed or finding a contractible edge  $xy$  with  $x \notin \{r, s_1, s_2\}$ .

- Suppose  $|V(G)| = 4$ . Then  $G = K_4$  and we can embed the lone vertex in  $V(G) - \{r, s_1, s_2\}$  anywhere in  $(A(s_1) - \{r\}) \cap (A(s_2) - \{r\})$ .
- Suppose  $|V(G)| = 5$ . By [1], there are at least  $\lceil 5/2 \rceil = 3$  contractible edges in  $G$  with equality only if  $G$  is 3-regular. But there are no 3-regular graphs of size 5, so there are at least 4 contractible edges, meaning there is at least one contractible edge  $xy$  with  $x \notin \{r, s_1, s_2\}$ .
- Suppose  $|V(G)| = 6$ . By [1], there are at least  $\lceil 6/2 \rceil = 3$  contractible edges in



$G$  with equality only if  $G$  is 3-regular. So, if there is no contractible edge  $xy$  with  $x \notin \{r, s_1, s_2\}$ , we may assume  $s_1 s_2 \in E(G)$  and  $G$  is 3-regular.

Then each of  $r, s_1, s_2$  has exactly one neighbor on outside of  $\{r, s_1, s_2\}$ , say  $v_r, v_{s_1}, v_{s_2}$  respectively. These three neighbors must be distinct, else they would form a cut. But each of  $v_r, v_{s_1}, v_{s_2}$  must have two more neighbors, hence they form a triangle. We now have the entire graph  $G$ , and the edge  $rv_r$  is contractible as desired.

- Suppose  $|V(G)| \geq 7$ . By [1] there are at least  $\lceil 7/2 \rceil = 4$  contractible edges in  $G$ , so there is at least one contractible edge  $xy$  with  $x \notin \{r, s_1, s_2\}$ .

In all cases where the embedding has not yet been found, we have a contractible edge  $xy$  with  $x \notin \{r, s_1, s_2\}$ .

Contract the edge  $xy$  to a vertex  $z$  and apply the induction hypothesis to embed  $G/xy$  in the 2-simplex, treating  $z$  as  $r, s_1$ , or  $s_2$  in  $G/xy$  if  $y$  is the corresponding vertex in  $G$ . We will embed  $G$  by placing one of  $x, y$  at the previous location of  $z$ , finding an appropriate new location for the other, and leaving all other vertices in their previous locations.

First, suppose  $y \notin \{r, s_1, s_2\}$  so that we have symmetry between  $x$  and  $y$ . Since  $z$  had neighbors embedded in each of  $A(z), B(z), C(z)$  by the induction hypothesis, and all neighbors of  $z$  are neighbors of either  $x$  or  $y$ , we can assume without loss of generality that  $y$  has neighbors embedded in each of  $A(z)$  and  $B(z)$  and  $x$  has a neighbor embedded in  $C(z)$ . We will embed  $y$  at the previous location of  $z$ , and find a location to embed  $x$  by applying Lemma 3.1 with  $p_0 = y$ ,  $Q = \mathcal{N}(x) - \{y\}$ , and  $\hat{q}$  any member of  $\mathcal{N}(x) \cap C(z)$ . The lemma assures us that:

1. The vertex  $x$  has neighbors in each of  $A(x), B(x), C(x)$ .
2. The vertex  $x$  occupies the same region as  $z$  with respect to any neighbors of  $x$ .
3. Since  $\hat{q} \in C(y), x \in C(y)$ .

Conditions 1 and 3 ensure that  $x$  and  $y$  have neighbors in the appropriate regions, and condition 2 ensures that no other vertices were affected by the new location of  $x$ . We have completed the case  $y \notin \{r, s_1, s_2\}$ .

If  $y \in \{r, s_1, s_2\}$ , then without loss of generality  $x$  has a neighbor in  $C(z)$  and we proceed as above. There are less conditions to check because  $y$  has fewer neighbor requirements. The proof is complete.  $\square$

*Proof of Conjecture 1.1 when  $k = 3$ .* The independent spanning trees  $T_1, T_2, T_3$  can be constructed as follows. Embed the vertices of  $G$  in a 2-simplex using Theorem 3.2. For each  $v \neq r$ , pick a neighbor in  $A(v)$  (excluding  $r$  if  $v \in \{s_1, s_2\}$ ) and add the corresponding edge to  $T_1$ . Similarly, for each  $v \notin \{r, s_1\}$  use a neighbor in  $B(v)$  to add an edge to  $T_2$ , and for each  $v \notin \{r, s_2\}$  use a neighbor in  $C(v)$  to add an edge to  $T_3$ . Finally, add  $rs_1$  to  $T_2$  and  $rs_2$  to  $T_3$ . We can confirm that these are spanning trees by noting that every vertex has degree at least one in each tree and then counting edges. These trees are independent since the paths back to  $r$  from each vertex travel through disjoint regions of the 2-simplex.  $\square$

### 3.3 Edge Version

We will prove the following:

**Theorem 3.3.** *Let  $G$  be a 3-edge-connected graph with  $r \in V(G)$  and  $e_1, e_2, e_3 \in E(G)$  distinct edges incident to  $r$  in  $G$ . Then there exists an embedding of  $E(G)$  in a 2-simplex with barycentric coordinates so that  $e_1 = (1, 0, 0), e_2 = (0, 1, 0), e_3 = (0, 0, 1)$ , and for every  $e \in E(G) - \{e_1, e_2, e_3\}$ ,  $e$  is adjacent in  $G$  to an edge in each of  $A(e), B(e)$ , and  $C(e)$ .*

*Proof.* We will proceed in a similar way as with Theorem 3.2, except we will add new edges to the embedding using the Mader construction instead of contraction.

We induct on  $|E(G)|$ . If  $|E(G)| = 4$ , we can embed the fourth edge anywhere to complete the proof. Otherwise, let  $G'$  be the graph immediately preceding  $G$  in a Mader

construction of  $G$ . Embed  $G'$  by induction, and consider the Mader operation needed to obtain  $G$  from  $G'$ .

First, suppose the operation is the addition of edge  $e$ . Let  $\hat{E}$  be the set of edges adjacent to  $e$  in  $G$ . Then we apply Lemma 3.1 to find a suitable location to embed  $e$ , with  $p_0$  an arbitrary member of  $\hat{E}$ ,  $Q = \hat{E} - \{p_0\}$ , and  $\hat{q}$  arbitrary. We only need condition 1 of the lemma, since the other edges already had their regions filled in  $G' = G - e$ .

Next, suppose the operation subdivides edge  $e$  into  $e_1$  and  $e_2$ , then adds a new edge  $e_3$  incident to the new vertex. Without loss of generality,  $e_1$  has adjacent edges embedded in each of  $A(e)$  and  $B(e)$  and  $e_2$  has an adjacent edge embedded in  $C(e)$ . We will embed  $e_1$  at the previous location of  $e$ , and find a location to embed  $e_2$  by applying Lemma 3.1 with  $p_0 = e_1$ ,  $Q$  the set of all edges adjacent to  $e_2$ , and  $\hat{q}$  any member of  $Q$  in  $C(e)$ . The lemma assures us that:

1. The edge  $e_2$  has adjacent edges in each of  $A(e_2)$ ,  $B(e_2)$ ,  $C(e_2)$ .
2. The edge  $e_2$  occupies the same region as  $e$  with respect to any edges adjacent to  $e_2$  (except  $e_1$ ).
3. Since  $\hat{q} \in C(e)$ ,  $e_2 \in C(e_1)$ .

Conditions 1 and 3 ensure that  $e_1$  and  $e_2$  have neighbors in the appropriate regions, and condition 2 ensures that no other vertices were affected by the new location of  $e_2$ . We can embed  $e_3$  as we did in the case of edge addition.

Finally, suppose the operation subdivides edge  $e$  into  $e_1$  and  $e_2$ , subdivides edge  $f$  into  $f_1$  and  $f_2$ , and then adds a new edge  $e_3$  incident to both new vertices. We proceed similarly to the previous case, embedding the subdivided pairs of edges the same way we embedded the subdivided pair previously. Again, we can embed  $e_3$  as we did in the case of edge addition. The proof is complete.

□

*Proof of Conjecture 1.2 when  $k = 3$ .* The edge-independent spanning trees  $T_1, T_2, T_3$  can be constructed as follows. Embed the edges of  $G$  in a 2-simplex using Theorem 3.3. Then, for each  $v \neq r$ , we will first find three incident edges  $e_1^v, e_2^v, e_3^v$  so that  $A(e_1^v)$ ,  $B(e_2^v)$ , and  $C(e_3^v)$  are disjoint.

To do this, apply Lemma 3.1 with  $p_0$  any edge incident to  $v$ ,  $Q$  the set of all other edges incident to  $v$ , and  $\hat{q}$  chosen arbitrarily. We obtain a point  $p$ , but we will not embed anything there or move any embedded objects there. Instead we use it only as a point of reference, observing by condition 1 of the lemma there are distinct edges incident to  $v$  in  $A(p), B(p), C(p)$ , which we call  $e_1^v, e_2^v, e_3^v$  respectively. Now, since  $e_1^v \in A(p)$ , we have  $A(e_1^v) \subset A(p)$ , and similarly  $B(e_2^v) \subset B(p)$  and  $C(e_3^v) \subset C(p)$ . Therefore  $A(e_1^v)$ ,  $B(e_2^v)$ , and  $C(e_3^v)$  are disjoint as desired.

We go further, finding edges  $\hat{e}_1^v, \hat{e}_2^v, \hat{e}_3^v$  with additional properties. If  $A(e_1^v)$  does not contain any edges incident to  $v$ , then set  $\hat{e}_1^v = e_1^v$ . Otherwise choose  $\hat{e}_1^v$  to be an edge in  $A(e_1^v)$  incident to  $v$  so that  $A(\hat{e}_1^v)$  does not contain any edges incident to  $v$ . Similarly define  $\hat{e}_2^v$  and  $\hat{e}_3^v$  using the  $B$  and  $C$  regions respectively. Note that  $A(\hat{e}_1^v)$ ,  $B(\hat{e}_2^v)$ , and  $C(\hat{e}_3^v)$  are disjoint, since they are subsets of  $A(e_1^v)$ ,  $B(e_2^v)$ , and  $C(e_3^v)$  respectively. Assign the edges  $\hat{e}_1^v, \hat{e}_2^v, \hat{e}_3^v$  to  $T_1, T_2, T_3$  respectively.

After doing this for each  $v \neq r$ , we have the desired trees. We can confirm that they are spanning trees by noting that every vertex has degree at least one in each tree and then counting edges.

As for edge-independence, consider a vertex  $v \neq r$  and its assigned edges  $\hat{e}_1^v, \hat{e}_2^v, \hat{e}_3^v$ . We will argue that the paths to  $r$  in each tree use edges from the disjoint sets  $A(\hat{e}_1^v)$ ,  $B(\hat{e}_2^v)$ , and  $C(\hat{e}_3^v)$  respectively.

If  $\hat{e}_1^v = e_1$  then trivially the  $T_1$ -path from  $v$  to  $r$  lies in  $A(\hat{e}_1^v)$ , so suppose  $\hat{e}_1^v \neq e_1$ . By Theorem 3.3,  $A(\hat{e}_1^v)$  is nonempty, but by the definition of  $\hat{e}_1^v$ , it does not contain any vertices incident to  $v$ . Thus,  $A(\hat{e}_1^v)$  contains an edge incident via the other endpoint of  $\hat{e}_1^v$ , say  $u$ . And by definition of  $\hat{e}_1^u$ , we know that  $\hat{e}_1^u \neq \hat{e}_1^v$ , so the subsequent edge in the  $T_1$ -path from

$v$  to  $r$  lies in  $A(\hat{e}_1^v)$ . But since  $v$  is arbitrary we can say the same for  $u$ , so the next edge in the path lies in  $A(\hat{e}_1^u) \subset A(\hat{e}_1^v)$ , and so on. Thus, the entire  $T_1$ -path from  $v$  to  $r$  lies in  $A(\hat{e}_1^v)$  as desired. Similar arguments show that the same is true for the  $T_2$ - and  $T_3$ -paths, so the trees are edge-independent. The proof is complete.  $\square$

### 3.4 Conjectures

We leave open the generalization of Theorems 3.2 and 3.3 to higher dimensions. The localization achieved by contraction and Mader operations seems to be insufficient to place new vertices and edges in higher dimensions.

Parameterize the  $(k - 1)$ -simplex as

$$S = \{(a_1, a_2, \dots, a_k) : 0 \leq a_1, a_2, \dots, a_k \leq 1 \text{ and } \sum_i a_i = 1\}.$$

Suppose  $p = (a_1^*, a_2^*, \dots, a_k^*) \in S$ . For  $i = 1, 2, \dots, k$ , define

$$A_i(p) = \{(a_1, a_2, \dots, a_k) \in S : a_i - a_i^* > a_j - a_j^* \text{ for all } j \neq i\}.$$

**Conjecture 3.4.** *Let  $G$  be a  $k$ -connected graph with  $r \in V(G)$  and  $s_2, s_3, \dots, s_k \in \mathcal{N}(r)$ . Then there exists an embedding of  $V(G)$  in a  $(k - 1)$ -simplex with barycentric coordinates so that:*

1.  $r = (1, 0, 0, \dots, 0)$ ,
2.  $s_2 = (0, 1, 0, 0, \dots, 0)$ ,  $s_3 = (0, 0, 1, 0, 0, \dots, 0)$ ,  $\dots$ ,  $s_k = (0, 0, \dots, 0, 1)$ ,
3. for every  $v \in V(G) - \{r, s_2, s_3, \dots, s_k\}$ ,  $v$  has a neighbor embedded in each of  $A_1(v), A_2(v), \dots, A_k(v)$ , and
4. for  $i = 2, 3, \dots, k$ , some vertex in  $\mathcal{N}(s_i) - \{r\}$  is embedded in  $A_j(s_i)$  for each  $j \neq i$ .

**Conjecture 3.5.** *Let  $G$  be a  $k$ -edge-connected graph with  $r \in V(G)$  and  $e_1, e_2, \dots, e_k \in E(G)$  distinct edges incident to  $r$  in  $G$ . Then there exists an embedding of  $E(G)$  in a  $(k - 1)$ -simplex with barycentric coordinates so that:*

1.  $e_1 = (1, 0, 0, \dots, 0), e_2 = (0, 1, 0, 0, \dots, 0), \dots, e_k = (0, 0, \dots, 0, 1)$ , and
2. *for every  $e \in E(G) - \{e_1, e_2, \dots, e_k\}$ ,  $e$  is adjacent in to an edge embedded in each of  $A_1(e), A_2(e), \dots, A_k(e)$ .*

## CHAPTER 4

### PATH-SWAPPING

Our goal is to approach the Vertex Conjecture from a different angle, by removing the “spanning” requirement on the trees. This allows us to induct on the number of vertices spanned by the trees, eventually reaching the original desired result. We begin by defining a generalization of independent trees:

**Definition.** Let  $G$  be a graph,  $r \in V(G)$ , and  $S \subseteq V(G)$ . Then we say two subtrees  $T_1, T_2$  are *independent with respect to*  $(r, S)$  if  $\{r\} \cup S \subseteq V(T_1) \cap V(T_2)$  and, for each  $s \in S$ , the unique path in  $T_1$  between  $r$  and  $s$  is internally disjoint from the unique path in  $T_2$  between  $r$  and  $s$ .

This definition extends pairwise to larger sets of trees.

We will prove the following:

**Proposition 4.1.** *Let  $r, s_1, s_2$  be distinct vertices of a planar graph  $G$ . Suppose that for  $i = 1, 2$ , there exist  $k$  internally vertex disjoint paths from  $r$  to  $s_i$ . Then  $G$  contains  $k$  independent subtrees with respect to  $(r, \{s_1, s_2\})$ .*

Let  $H$  be a subgraph of  $G$  such that  $H$  satisfies the hypothesis of Proposition 4.1 and  $|E(H)|$  is minimal. It suffices to prove the result in  $H$ . We induct on  $k$ ; the base case  $k = 1$  is trivial. Denote the disjoint paths from  $r$  to  $s_1$  by  $P_1, \dots, P_k$  and the disjoint paths from  $r$  to  $s_2$  by  $Q_1, \dots, Q_k$ .

**Claim 6.** *At least one of the following is true:*

1. *For every  $i = 1, 2, \dots, k$ ,  $P_i$  has an internal intersection with one of the  $Q$  paths.*
2. *For every  $i = 1, 2, \dots, k$ ,  $Q_i$  has an internal intersection with one of the  $P$  paths.*

3. The conclusion of Proposition 4.1 is true.

*Proof.* Suppose 1 and 2 are false, with  $P_i$  and  $Q_j$  as respective counterexamples. Then  $P_i \cup Q_j$  is a tree and only intersects the rest of  $H$  at  $r, s_1, s_2$ . Apply the induction hypothesis to  $H - (P_i \cup Q_j - \{r, s_1, s_2\})$  to construct  $k-1$  independent trees with respect to  $(r, \{s_1, s_2\})$ , and use  $P_i \cup Q_j$  as the final tree. It is independent from the others because it is completely disjoint from them except at  $r, s_1, s_2$ .  $\square$

Without loss of generality, we can assume in the remainder of the proof that item 1 is true. That is, for every  $i = 1, 2, \dots, k$ ,  $P_i$  has an internal intersection with one of the  $Q$  paths.

**Claim 7.** For every  $i = 1, 2, \dots, k$ , the edge of  $P_i$  incident to  $r$  is in  $E(Q_j)$  for some  $j$ .

**Proof.** Suppose not. Then there is some subpath  $rP_iv$  which does not intersect any of the  $Q$  paths internally. But  $P_i$  must intersect a  $Q$  path internally by Claim 6, so if we choose the subpath to be maximal, then  $v$  is in some  $Q_j$ . Consider the subgraph  $H'$  obtained by deleting any edges in  $rQ_jv$  which are not used by paths other than  $Q_j$ . We can see that this deletion is nontrivial by following  $rQ_jv$  from  $v$  towards  $r$ . We start at  $v \in V(P_i)$ , and at some point must leave  $P_i$  (else  $P_i$  would contain a cycle through  $r$ ). The first edge we reach that is not in  $P_i$  has an endpoint in  $P_i$ , so it cannot be part of any other  $P$  path and will be deleted. In  $H'$ , we can replace  $Q_j$  with  $rP_iv \cup vQ_js_2$ , which is disjoint from the other  $Q$  paths by the choice of  $rP_iv$ . This contradicts the minimality of  $|E(H)|$ .  $\square$

By permuting indices, we can assume for convenience that  $P_1$  is coincident to  $Q_1$  near  $r$ ,  $P_2$  is coincident to  $Q_2$  near  $r$ , and so on.

For  $i = 1, 2, \dots, k$ , we make several definitions. Follow  $P_i$  from  $r$  towards  $s_1$ . We know that  $P_i$  is coincident with  $Q_i$  at the start of this process, so they will eventually split. Denote the site of this split by  $v_i$ , noting that  $v_i$  could be  $s_1$  or  $s_2$ . After the split, if  $v_i \neq s_1$ , continue following  $P_i$  towards  $s_1$  until the next (not necessarily internal) intersection with a  $Q$  path. If such an intersection exists, denote the site by  $w_i$  and the index of the intercepted



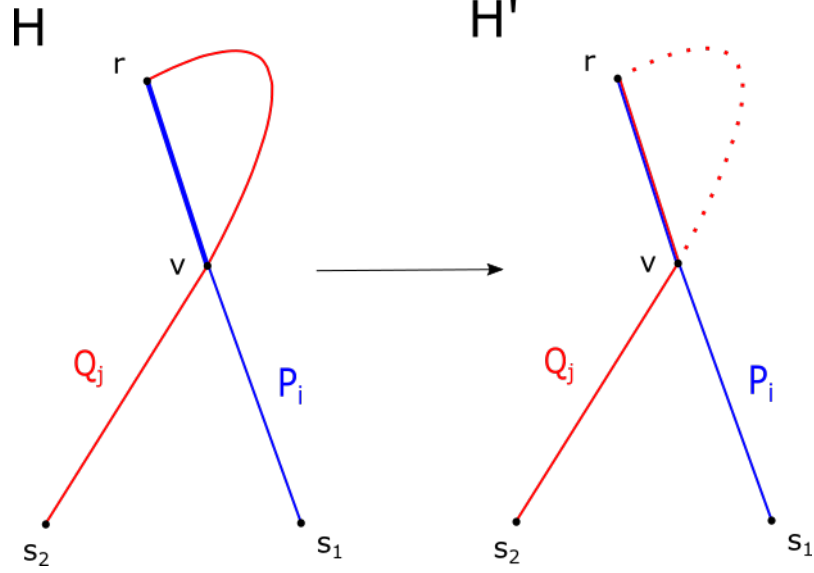


Figure 4.1: An illustration of the proof of Claim 7. In  $H$ , the bold segment does not intersect any  $Q$  paths internally. In  $H'$ , edges in the dashed segment which are not used by any  $P$  path are deleted.

$Q$  path by  $n_i$  (chosen arbitrarily if  $w_i = s_2$ ). If no such interception exists (or  $v_i = s_1$ ), then  $w_i$  and  $n_i$  are undefined.

**Claim 8.** *There is no sequence of distinct indices  $i_1, i_2, \dots, i_m$ , where  $1 \leq m \leq k$ , so that  $n_{i_j} = i_{j+1}$  for  $j = 1, 2, \dots, m-1$  and  $n_{i_m} = i_1$ .*

*Proof.* Suppose there is such a sequence. By permuting indices, we can assume for convenience that the sequence is  $i_1 = 1, i_2 = 2, \dots, i_m = m$  so that  $n_j = j+1$  for  $j = 1, 2, \dots, m-1$  and  $n_m = 1$ . We reroute the paths  $Q_1, Q_2, \dots, Q_m$  to contradict edge minimality as follows. Consider the subgraph  $H'$  obtained by deleting any edges in  $v_j Q_j w_{j-1}$  which are not used by any other path, for  $j = 1, 2, 3, \dots, m$ , where  $w_0 = w_m$ . We know that this deletion is nontrivial since the  $v_1 \in V(P_1) \cap V(Q_1)$ , so the edge incident to  $v_1$  in  $v_1 Q_1 s_2$  cannot be part of any other path.

In  $H'$ , we replace  $Q_j$  with  $r P_{j-1} w_{j-1} \cup w_{j-1} Q_j s_2$  for  $j = 1, 2, 3, \dots, m$ , where  $P_0 = P_m$  and  $w_0 = w_m$ . These new  $Q$  paths are disjoint from each other and the other  $Q$  paths, since each  $r P_{j-1} v_{j-1}$  was vacated by the reroute and each  $v_{j-1} P_{j-1} w_{j-1}$  does not intersect any  $Q$  paths internally by definition. This contradicts the minimality of  $|E(H)|$ .  $\square$

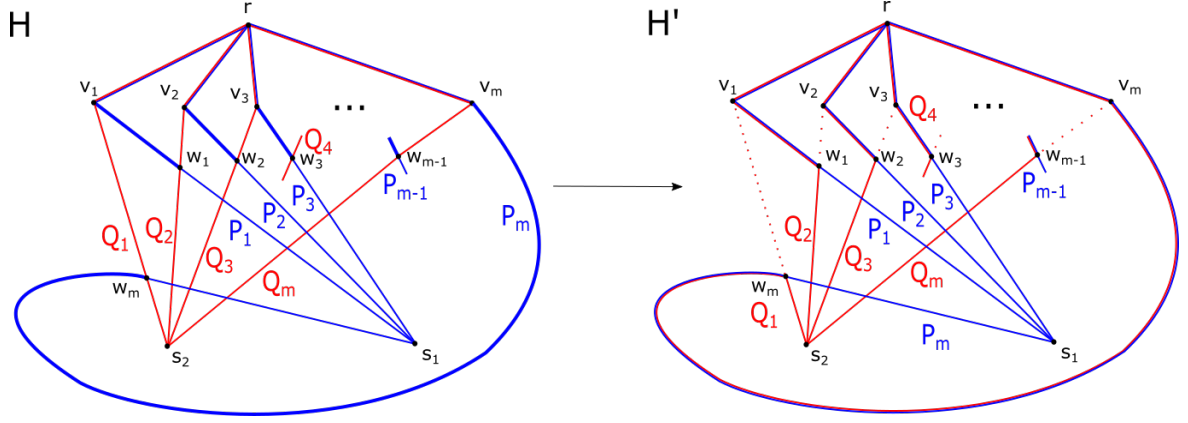


Figure 4.2: An illustration of the proof of Claim 8. In  $H$ , the bold segments do not intersect any  $Q$  paths internally. In  $H'$ , edges in the dashed segments which are not used by any  $P$  path are deleted.

Next, we make a few observations about the  $n_i$  indices:

- Using Claim 8 with  $m = 1$ , we know that  $n_i \neq i$  for all  $i$ .
- If every  $n_i$  is defined, we contradict Claim 8 with a pigeonhole argument.
- If every  $n_i$  is undefined, then each of  $P_1 \cup Q_1, P_2 \cup Q_2, \dots, P_k \cup Q_k$  is a tree, and so the proof is complete.

So, we can assume that some  $n_i$  is defined and another is undefined. By permuting indices, we can assume  $n_1$  is defined with  $n_1 = 2$ . If  $n_2$  is defined, then  $n_2 \neq 1$  by Claim 8, so we can permute indices to assume  $n_2 = 3$ . Continuing in this fashion, we must eventually reach an undefined  $n_a$ .

Now consider following  $P_1$  from  $s_1$  towards  $r$  until the first intersection with one of the  $Q$  paths. We know that this will happen before we reach  $v_1$ , because  $P_1$  intersects  $Q_2$  at  $w_1$ . Say this intersection is with some  $Q_b$  at a vertex  $x$ .

**Claim 9.**  $b > a$ .

*Proof.* Suppose  $b \leq a$ . We will reroute the paths  $Q_b, Q_{b+1}, \dots, Q_a$ , similar to the proof of Claim 8. Consider the subgraph  $H'$  obtained by deleting any edges in  $v_i Q_i w_{i-1}$  which are

not used by any other path, for  $i = b + 1, b + 2, \dots, a$ , and by deleting any edges in  $v_b Q_b x$  which are not used by any other path. We know that this deletion is nontrivial since the  $v_b \in V(P_b) \cap V(Q_b)$ , so the edge incident to  $v_b$  in  $v_b Q_b s_2$  cannot be part of any other path.

In  $H'$ , for each  $i = b + 1, b + 2, \dots, a$ , we replace  $Q_i$  with  $r P_{i-1} w_{i-1} \cup w_{i-1} Q_i s_2$ . We replace  $Q_b$  by  $P_a \cup s_1 P_1 x \cup x Q_b s_2$ . These new  $Q$  paths are disjoint from each other and the other  $Q$  paths since each  $r P_{i-1} v_{i-1}$  (and  $r P_a v_a$ ) was vacated by the reroute, each  $v_{i-1} P_{i-1} w_{i-1}$  is internally disjoint from the  $Q$  paths definition of  $w_{i-1}$ ,  $v_a P_a s_1$  is internally disjoint from the  $Q$  paths since  $n_a$  is undefined, and  $s_1 P_1 x$  is internally disjoint from the  $Q$  paths by the definition of  $x$ . This contradicts the minimality of  $|E(H)|$ .  $\square$

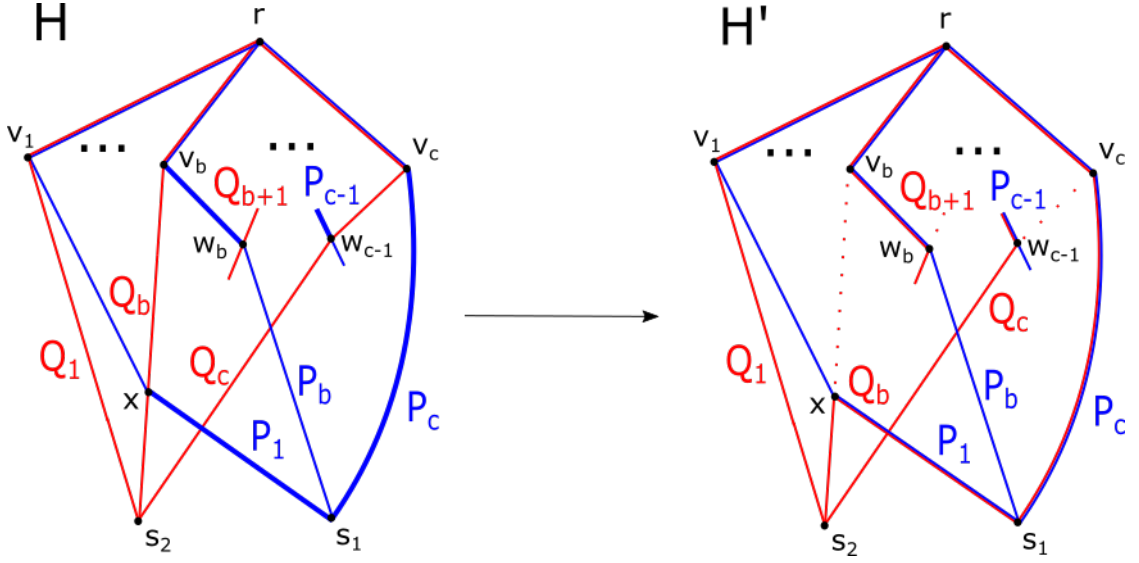


Figure 4.3: An illustration of the proof of Claim 9. In  $H$ , the bold segments do not intersect any  $Q$  paths internally. In  $H'$ , edges in the dashed segments which are not used by any  $P$  path are deleted.

Since  $b > a$ , we can permute indices so that  $b = a + 1$ . Consider  $n_{a+1}$ . If it is defined and  $n_{a+1} > a + 1$ , we can permute indices so  $n_{a+1} = a + 2$  and consider  $n_{a+2}$ . Eventually, we will reach an  $n_c$  which is either smaller than  $c$  or undefined. We address these two cases separately:

- Suppose  $n_c < c$ . If  $n_c > a$ , we have contradicted Claim 8 with indices  $n_c, n_c + 1, \dots, c$ . So we may assume  $n_c \leq a$ . We will reroute the paths  $Q_{n_c}, Q_{n_c+1}, \dots, Q_c$ ,

similar to the proof of Claim 8. Consider the subgraph  $H'$  obtained by deleting any edges in  $v_i Q_i w_{i-1}$  which are not used by any other path, for  $i = n_c + 1, n_c + 2, \dots, c, i \neq a + 1$ , and also deleting any edges in  $v_{n_c} Q_{n_c} w_c$  and  $v_{a+1} Q_{a+1} x$  which are not used by any other path. We know that this deletion is nontrivial since the  $v_b \in V(P_b) \cap V(Q_b)$ , so the edge incident to  $v_b$  in  $v_b Q_b s_2$  cannot be part of any other path.

In  $H'$ , we replace each  $Q_i$  with  $rP_{i-1}w_{i-1} \cup w_{i-1}Q_i s_2$  for  $i = n_c + 1, n_c + 2, \dots, c, i \neq a + 1$ , we replace  $Q_{n_c}$  with  $rP_c w_c \cup w_c Q_{n_c} s_2$ , and we replace  $Q_{a+1}$  with  $P_a \cup s_1 P_1 x \cup x Q_{a+1} s_2$ . These new  $Q$  paths are disjoint from each other and the other  $Q$  paths since each  $rP_{i-1}v_{i-1}$  (and  $rP_c v_c, rP_a v_a$ ) was vacated by the reroute, each  $v_{i-1}P_{i-1}w_{i-1}$  (and  $v_c P_c w_c$ ) is internally disjoint from the  $Q$  paths by definition of  $w_{i-1}$ ,  $v_a P_a s_1$  is internally disjoint from the  $Q$  paths since  $n_a$  is undefined, and  $s_1 P_1 x$  is internally disjoint from the  $Q$  paths by the definition of  $x$ . This contradicts the minimality of  $|E(H)|$ .  $\square$

- Suppose  $n_c$  is undefined. We will reroute the paths  $Q_{a+1}, Q_{a+2}, \dots, Q_c$ , similar to the proof of Claim 8. Consider the subgraph  $H'$  obtained by deleting any edges in  $v_i Q_i w_{i-1}$  which are not used by any other path, for  $i = a + 2, a + 3, \dots, c$ , and also deleting any edges in  $v_{a+1} Q_{a+1} x$  which are not used by any other path. We know that this deletion is nontrivial since the  $v_b \in V(P_b) \cap V(Q_b)$ , so the edge incident to  $v_b$  in  $v_b Q_b s_2$  cannot be part of any other path.

In  $H'$ , for each  $i = a + 2, a + 3, \dots, c$ , we replace  $Q_i$  with  $rP_{i-1}w_{i-1} \cup w_{i-1}Q_i s_2$ . We can replace  $Q_{a+1}$  by  $P_c \cup s_1 P_1 x \cup x Q_{a+1} s_2$ . These new  $Q$  paths are disjoint from each other and the other  $Q$  paths since each  $rP_{i-1}v_{i-1}$  (and  $rP_c v_c$ ) was vacated by the reroute, each  $v_{i-1}P_{i-1}w_{i-1}$  is internally disjoint from the  $Q$  paths by the definition of  $w_{i-1}$ ,  $v_c P_c s_1$  is internally disjoint from the  $Q$  paths since  $n_c$  is undefined, and  $s_1 P_1 x$  is internally disjoint from the  $Q$  paths by the definition of  $x$ . This contradicts the

minimality of  $|E(H)|$ .

Both remaining cases led to contradictions. The proof of Proposition 4.1 is complete.

□

## CHAPTER 5

### THE GYÖRI-LOVÁSZ THEOREM

#### 5.1 Original Theorem

Our objective is to give a self-contained proof of Theorem 1.4. The proof we give is Györi's original proof, restated using new terminology. It clearly suffices to prove the following.

**Theorem 5.1.** *Let  $k \geq 2$  be an integer, let  $G$  be a  $k$ -connected graph on  $n$  vertices, let  $v_1, v_2, \dots, v_k$  be distinct vertices of  $G$ , and let  $n_1, n_2, \dots, n_k$  be positive integers with  $n_1 + n_2 + \dots + n_k < n$ . Let  $G_1, G_2, \dots, G_k$  be disjoint connected subgraphs of  $G$  such that, for  $i = 1, 2, \dots, k$ , the graph  $G_i$  has  $n_i$  vertices and  $v_i \in V(G_i)$ . Then  $G$  has disjoint connected subgraphs  $G'_1, G'_2, \dots, G'_k$  such that  $v_i \in V(G'_i)$  for  $i = 1, 2, \dots, k$ , the graph  $G'_1$  has  $n_1 + 1$  vertices and for  $i = 2, 3, \dots, k$  the graph  $G'_i$  has  $n_i$  vertices.*

##### 5.1.1 Terminology and Definitions

For the proof of Theorem 5.1 we will use terminology inspired by hydrology. Certain vertices will act as “dams” by blocking other vertices from the rest of a subgraph of  $G$ , thus creating a “reservoir”. A sequence of dams will be called a “cascade”.

To define these notions precisely let  $G_1, G_2, \dots, G_k$  be as in Theorem 5.1 and let  $i = 2, 3, \dots, k$ . For a vertex  $v \in V(G_i)$  we define the **reservoir** of  $v$ , denoted by  $R(v)$ , to be the set of all vertices in  $G_i$  which are connected to  $v_i$  by a path in  $G_i \setminus v$ . Note that  $v \notin R(v)$  and also  $R(v_i) = \emptyset$ . By a **cascade** in  $G_i$  we mean a (possibly null) sequence  $w_1, w_2, \dots, w_m$  of distinct vertices in  $G_i \setminus v_i$  such that  $w_{j+1} \notin R(w_j)$  for  $j = 1, \dots, m - 1$ . Thus  $w_j$  separates  $w_{j-1}$  from  $w_{j+1}$  in  $G_i$  for every  $j = 1, \dots, m - 1$ , where  $w_0$  means  $v_i$ . By a **configuration** we mean a choice of subgraphs  $G_1, G_2, \dots, G_k$  as in Theorem 5.1 and exactly one cascade in each  $G_i$  for  $i = 2, 3, \dots, k$ . By a **cascade vertex** we mean a vertex

belonging to one of the cascades in the configuration. We define the **rank** of some cascade vertices recursively as follows. Let  $w \in V(G_i)$  be a cascade vertex. If  $w$  has a neighbor in  $G_1$ , then we define the rank of  $w$  to be 1. Otherwise, its rank is the least integer  $k \geq 2$  such that there is a cascade vertex  $w' \in V(G_j)$ , for some  $j \in \{2, 3, \dots, k\} - \{i\}$ , so that  $w$  has a neighbor in  $R(w')$  and  $w'$  has rank  $k - 1$ . If there is no such neighbor, then the rank of  $w$  is undefined. For an integer  $r \geq 1$ , let  $\rho_r$  denote the total number of vertices belonging to  $R(w)$  for some cascade vertex  $w$  of rank  $r$ . A configuration is **valid** if each cascade vertex has well-defined rank and this rank is strictly increasing within a cascade. That is, for each cascade  $w_1, w_2, \dots, w_m$  and integers  $1 \leq i < j \leq m$  the rank of  $w_i$  is strictly smaller than the rank of  $w_j$ . Note that a valid configuration exists trivially by taking each cascade to be the null sequence. For an integer  $r \geq 1$  a valid configuration is  **$r$ -optimal** if, among all valid configurations, it maximizes  $\rho_1$ , subject to that it maximizes  $\rho_2$ , and so on, up to maximizing  $\rho_r$ . If a valid configuration is  $r$ -optimal for all  $r \geq 1$ , we simply say it is **optimal**.

Finally, we define  $S := V(G) - V(G_1) - V(G_2) - \dots - V(G_k)$ . This is nonempty in the setup of Theorem 2. We say that a **bridge** is an edge with one end in  $S$  and the other end in the reservoir of a cascade vertex. In a valid configuration, the **rank** of the bridge is the minimum rank of all cascade vertices  $w$  where the bridge has an end in  $R(w)$ .

These concepts are illustrated in Figure 5.1.

### 5.1.2 Lemmas

**Lemma 5.2.** *If there is an optimal configuration containing a bridge, then the conclusion of Theorem 5.1 holds.*

*Proof.* Suppose there is an optimal configuration containing a bridge. Then for some  $r \in \mathbb{N}$  we can find a configuration which is  $r$ -optimal containing a bridge of rank  $r$ . Choose the configuration and bridge so that  $r$  is minimal. Denote the endpoints of the bridge as  $a \in S$  and  $b \in R(w) \subseteq V(G_i)$ , where  $w$  is a cascade vertex of rank  $r$ .

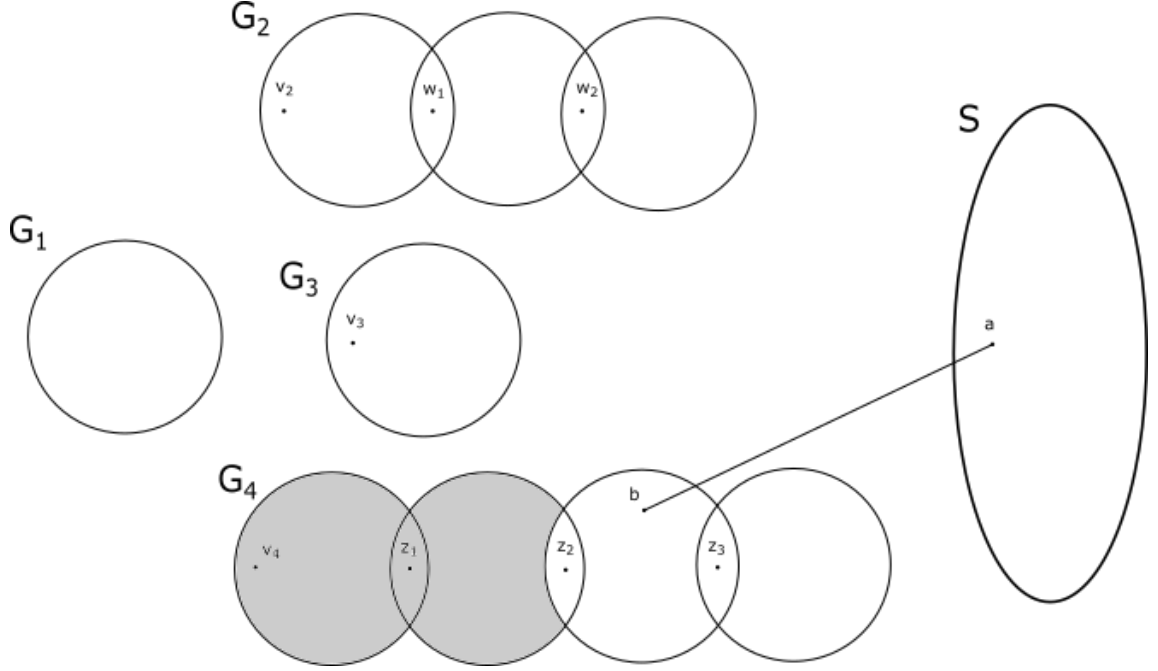


Figure 5.1: An example of a configuration.  $w_1, w_2, z_1, z_2$ , and  $z_3$  are cascade vertices.  $R(z_2)$  is shaded. The edge  $ab$  is a bridge, and its rank is the rank of  $z_3$ .

Suppose  $w$  separates  $G_i$ . Since we have a valid configuration, any cascade vertices in  $V(G_i) - R(w) - \{w\}$  must have rank greater than  $r$ . Choose any nonseparating vertex from this set, say  $u$ . We make a new valid configuration in the following way. Move  $u$  to  $S$  and  $a$  to  $G_i$ . Leave the cascades the same with one exception: remove all cascade vertices in  $V(G_i) - R(w) - \{w\}$  and all cascade vertices whose rank becomes undefined. Note that any cascade vertices affected by this action have rank greater than  $r$ . Now our new configuration is valid, increased the size of  $R(w)$ , and did not change any other reservoirs of rank at most  $r$ . This contradicts  $r$ -optimality.

So, continue under the assumption that  $w$  does not separate  $G_i$ . If  $r = 1$ , choose  $G'_1 := G_1 + w$ , the graph obtained from  $G_1$  by adding the vertex  $w$  and all edges from  $w$  to  $G_1$ ,  $G'_i := (G_i + a) \setminus w$ , and leave all other  $G_j$ 's unchanged. Then these graphs satisfy the conclusion of Theorem 5.1, as desired.

If  $r > 1$ , then  $w$  has a neighbor in some  $R(w')$  with  $\text{rank}(w') = r - 1$ . As before, we make a new valid configuration by moving  $w$  to  $S$  and  $a$  to  $G_i$ . Keep the cascades the



same as before, except terminate  $w$ 's former cascade just before  $w$  and exclude any cascade vertices whose rank has become undefined. Though we may have lost several reservoirs of rank  $r$  and above, the new configuration is still  $(r - 1)$ -optimal. Also, the edge connecting  $w$  to its neighbor in  $R(w')$  is now a rank  $r - 1$  bridge. This contradicts the minimality of  $r$ , so the proof of Lemma 5.2 is complete.  $\square$

**Lemma 5.3.** *Suppose there is an optimal configuration with an edge  $ab$  such that:*

1. *Either  $a \in V(G_1)$  or  $a$  is in a reservoir, and*
2.  *$b \in V(G_i)$  for some  $i \in \{2, 3, \dots, k\}$ ,  $b \neq v_i$ , and  $b$  is not in a reservoir.*

*Then the cascade of  $G_i$  is not null and  $b$  is the last vertex in the cascade.*

*Proof.* Suppose there is such an edge in an optimal configuration and  $b$  is not the last vertex in the cascade of  $G_i$ . Denote the cascade of  $G_i$  by  $w_1, \dots, w_m$  (which a priori could be null). Since  $b$  is not in a reservoir and is not the last cascade vertex, we know that  $b$  is not a cascade vertex. Then make a new configuration by including  $b$  at the end of  $G_i$ 's cascade. By condition 1,  $b$  has well-defined rank. If this rank is larger than all other ranks in the cascade (including the case where the former cascade is null), then we have a valid configuration and have contradicted optimality by adding a new reservoir (which is nonempty since  $v_i \in R(b)$ ) without changing anything else.

So, the former cascade is not null. Let  $\text{rank}(b) = r$  and let  $j \geq 0$  be the integer such that  $j = 0$  if  $r \leq \text{rank}(w_1)$  and  $\text{rank}(w_j) < r \leq \text{rank}(w_{j+1})$  otherwise. We make a second adjustment by excluding the vertices  $w_{j+1}, w_{j+2}, \dots, w_m$  from the cascade and adding  $b$  to it. Now the configuration is clearly valid, but it is unclear whether optimality has been contradicted. But notice that every vertex which used to belong to  $R(w_{j+1}) \cup R(w_{j+2}) \cup \dots \cup R(w_m)$  now belongs to  $R(b)$ , and also  $R(b)$  contains  $w_m$  which was not in any reservoir previously. Thus, we have strictly increased the size of rank  $r$  reservoirs without affecting any lower rank reservoirs. This contradicts optimality, so the proof of Lemma 5.3 is complete.  $\square$

### 5.1.3 Proof of Theorem 5.1

Using our lemmas, we can assume we have an optimal configuration which does not contain any bridges and where any edges as in Lemma 5.3 are at the end of their cascades. Consider the set containing the last vertex in each non-null cascade and the  $v_i$  corresponding to each null cascade. This is a cut of size  $k - 1$ , separating  $G_1$  and the reservoirs from the rest of the graph, including  $S$ . This contradicts  $k$ -connectivity, and the proof is complete.  $\square$

### 5.1.4 Algorithm

**Theorem 5.4.** *There exists an algorithm with the following specifications:*

**Input:** *A  $k$ -connected graph  $G$  on  $n$  vertices, distinct vertices  $v_1, v_2, \dots, v_k$  of  $G$ , and natural numbers  $n_1, n_2, \dots, n_k$  such that  $n_1 + n_2 + \dots + n_k = n$ .*

**Output:** *A partition of  $V(G)$  into  $V_1, V_2, \dots, V_k$ , such that, for  $i = 1, 2, \dots, k$ ,  $G[V_i]$  is connected,  $v_i \in V_i$  and  $|V_i| = n_i$ .*

**Running time:**  $O^* \left( 2^{2(1-\frac{1}{k})n} \right)$

*Proof.* Initialize  $V_i = \{v_i\}$  for  $i = 1, 2, \dots, k$ . The algorithm will have  $n - k$  iterations, growing  $\sum_i |V_i|$  by one in each iteration.

Begin each iteration by permuting indices so that  $|V_1| < n_1$  and  $|V_1|$  is minimal among all  $i$  with  $|V_i| < n_i$  (breaking ties arbitrarily). Initialize a configuration with null sequences for every cascade. The iteration will have several subiterations, each time either improving the optimality of the configuration, or growing  $|V_1|$  and therefore terminating the iteration. The process for each subiteration is as follows.

- Search for an edge between  $V_1$  and  $S$ . If one is found, move the endpoint in  $S$  to  $V_1$ . We have grown  $|V_1|$ , so move on to the next iteration.
- Search for an edge  $ab$  so that  $a$  is in a reservoir,  $b \in V_i$  for some  $i \neq 1$ ,  $b \neq v_i$ , and  $b$  is not in a reservoir. If one is found, define  $b$  as a new cascade vertex. If its rank is

lower than other cascade vertices in  $V_i$ , exclude those vertices from the cascade. We have improved optimality, so move on to the next subiteration.

- Search for a bridge. When found,
  - Denote the endpoints of the bridge as  $a \in S$  and  $b \in R(w) \subseteq V(G_i)$ , where  $w$  is a cascade vertex of rank  $r$ .
  - If  $w$  separates  $G_i$ , find a nonseparating vertex  $u$  in  $V(G_i) - R(w) - \{w\}$ . Move  $u$  to  $S$  and  $a$  to  $G_i$ . Remove all cascade vertices in  $V(G_i) - R(w) - \{w\}$  and all cascade vertices whose rank becomes undefined. We have improved optimality, so move on to the next subiteration.
  - If  $w$  does not separate  $G_i$ ,
    - \* If  $r = 1$ , add  $w$  to  $V_1$ . We have grown  $|V_1|$ , so move on to the next iteration.
    - \* If  $r > 1$ , then  $w$  has a neighbor with lower rank. Move  $w$  to  $S$  and  $a$  to  $G_i$ . Keep the cascades the same as before, except terminate  $w$ 's former cascade just before  $w$  and exclude any cascade vertices whose rank has become undefined. The edge between  $w$  and its lower rank neighbor is now a bridge. Repeat the above steps with the new bridge. As the rank of the bridge has strictly decreased, this loop will terminate in polynomial time.

The steps for each subiteration can be completed in polynomial time. We need to analyze the maximum number of subiterations.

We will argue that there are most  $n(1 - \frac{1}{k})$  reservoir vertices. If  $|V_1| \geq n/k$  this is trivial, so assume  $|V_1| < n/k$ . Recall that we chose  $V_1$  so that  $|V_1|$  is minimal among all  $i$  with  $|V_i| < n_i$ , and note that the algorithm increases some  $|V_i|$  by exactly 1 in each iteration. Therefore no other  $V_i$  can be larger than  $|V_1| + 1$ , as we would have chosen to grow  $V_1$  before any subgraph of size  $|V_1| + 1$ . Then  $|V_i| \leq |V_1| + 1 < n/k + 1$  for all  $i$ , which means there are at most  $(n/k)(k - 1) = n(1 - \frac{1}{k})$  reservoir vertices, as desired.

With this bound on the number of reservoir vertices, we can bound the number of possible sequences  $(\rho_1, \rho_2, \dots)$ . Since each subiteration increases this sequence lexicographically, we will be bounding the number of subiterations. For brevity, write  $c = (1 - \frac{1}{k})$ . Since  $\sum_r \rho_r \leq cn$ , the number of sequences is bounded by

$$\binom{2cn}{cn} = \frac{(2cn)!}{((cn)!)^2} \sim \frac{\sqrt{2\pi(2cn)} \left(\frac{2cn}{e}\right)^{2cn}}{\left(\sqrt{2\pi(cn)} \left(\frac{cn}{e}\right)^{cn}\right)^2} = \frac{2^{2cn}}{\sqrt{\pi cn}}$$

So we have at most  $2^{2(1-\frac{1}{k})n}$  subiterations, which becomes the limiting factor to the algorithm's runtime.  $\square$

## 5.2 The Generalized Györi-Lovász Theorem

Next, we consider generalizations of Theorem 1.4. Chen et al. extended this result to weighted, directed graphs by using flows [3]. Chandran, Cheung, and Isaac generalized Györi's original proof to integer-weighted, undirected graphs [2]. We will combine these results, using Chandran, Cheung, and Isaac's generalization of Györi's proof to prove a slight strengthening of the weighted, directed graph generalization presented by Chen et al.

The generalization is as follows. Let  $G$  be a directed graph on  $n$  vertices with  $v_1, v_2, \dots, v_k \in V(G)$ . Suppose  $G$  is  $k$ -connected to  $\{v_1, v_2, \dots, v_k\}$  in the directed sense, that is, for every  $v \notin \{v_1, v_2, \dots, v_k\}$ , there is a set of  $k$  directed paths from  $v$  to each of  $v_1, v_2, \dots, v_k$  which are pairwise vertex-disjoint (except at  $v$ ). We say that  $G$  is connected to  $v$  if it is 1-connected to  $\{v\}$ . Let  $w : V(G) \rightarrow \mathbb{R}^+$  be a weight function with the notation  $w(S) = \sum_{s \in S} w(s)$ . Let  $T_1, T_2, \dots, T_k \in \mathbb{R}^+$ , where  $T_i \geq w(v_i)$  and  $\sum T_i = w(V(G))$ . Finally, define  $w_{\max} = \max_{v \in V(G) - \{v_1, v_2, \dots, v_k\}} w(v)$ .

**Theorem 5.5.** *There exists a partitioning of  $V(G)$  into  $k$  parts  $V_1, V_2, \dots, V_k$  so that for each  $i = 1, 2, \dots, k$ ,  $v_i \in V_i$ ,  $G[V_i]$  is connected to  $v_i$ , and  $T_i - w_{\max} < w(V_i) < T_i + w_{\max}$ .*

Note that the original result is obtained by replacing each undirected edge with two

directed edges and taking  $w \equiv 1$ . In fact, we get something slightly stronger than the original result because we only require the graph to be  $k$ -connected to  $\{v_1, v_2, \dots, v_k\}$ , rather than general  $k$ -connectivity. We will prove Theorem 5.5 by proving the following.

**Theorem 5.6.** *Suppose we have disjoint sets  $W_1, W_2, \dots, W_k \subset V(G)$  so that for each  $i = 1, 2, \dots, k$ ,  $v_i \in W_i$  and  $G[W_i]$  is connected to  $v_i$ . Suppose further that  $S := V(G) - W_1 - W_2 - \dots - W_k$  is nonempty. To each  $W_i$ , assign a  $t_i \in \mathbb{R}$ .*

1. *If  $w(W_i) < t_i + w_{\max}$  for each  $i$  and  $w(W_1) < t_1$ , then there exist disjoint sets  $V_1, V_2, \dots, V_k \subset V(G)$  so that for each  $i = 1, 2, \dots, k$ ,  $G[V_i]$  is connected to  $v_i \in V_i$ ,  $w(V_i) < t_i + w_{\max}$ ,  $w(V_1) > w(W_1)$ , and each  $w(V_i) \geq w(W_i)$  unless  $w(W_i) \geq t_i$ , in which case we maintain  $w(V_i) \geq t_i$ .*
2. *If  $w(W_i) \leq t_i + w_{\max}$  for each  $i$  and  $w(W_1) \leq t_1$ , then there exist disjoint sets  $V_1, V_2, \dots, V_k \subset V(G)$  so that for each  $i = 1, 2, \dots, k$ ,  $G[V_i]$  is connected to  $v_i \in V_i$ ,  $w(V_i) \leq t_i + w_{\max}$ ,  $w(V_1) > w(W_1)$ , and each  $w(V_i) \geq w(W_i)$  unless  $w(W_i) > t_i$ , in which case we maintain  $w(V_i) > t_i$ .*

### 5.2.1 Terminology and Definitions

We will define terminology to merge the two cases of Theorem 5.6. Let  $W$  be a vertex set containing  $v_i$  and no other  $v_j$ . In case 1, we will say that  $W$  is *underweight* if  $w(W) < t_i$ ,  $W$  is *good* if  $t_i \leq w(W) < t_i + w_{\max}$ , or  $W$  is *overweight* if  $w(W) \geq t_i + w_{\max}$ . In case 2, we will say that  $W$  is *underweight* if  $w(W) \leq t_i$ ,  $W$  is *good* if  $t_i < w(W) \leq t_i + w_{\max}$ , or  $W$  is *overweight* if  $w(W) > t_i + w_{\max}$ . Note that, in both cases, any single vertex (besides a  $v_i$ ) can be added to an underweight vertex set, and the resulting vertex set will not be overweight. Similarly, any single vertex (besides a  $v_i$ ) can be removed from an overweight vertex set, and the resulting vertex set will not be underweight.

To prove Theorem 5.5 using Theorem 5.6, we initialize  $W_i := \{v_i\}$  and repeatedly apply case 2 of Theorem 5.6 with  $t_i = T_i - w_{\max}$ . Since there are a finite number of

weight combinations, repeatedly growing the weight of the first subgraph eventually makes it good, at which point we permute indices to call a new underweight subgraph  $W_1$ . Since Theorem 5.6 maintains the goodness of good vertex sets, the process will terminate. Since we maintain  $w(W_i) \leq t_i + w_{\max} = T_i$  and  $\sum T_i = w(V(G))$ ,  $S$  can only become empty if  $w(W_i) = T_i$  for each  $i$ , at which point we have the result of Theorem 5.5. So we continue under the assumption that  $S$  is nonempty when the process terminates. Then the process must have terminated because all sets are good. The choice of  $t_i = T_i - w_{\max}$  gives us the desired lower bound for Theorem 5.5,  $T_i - w_{\max} < w(W_i)$ .

Next, we continue to apply Theorem 5.6, but now using case 1 with  $t_i = T_i$ . Again, the process will terminate for the same reasons. This time, however, since good sets have  $w(W_i) \geq t_i = T_i$  and  $\sum T_i = w(V(G))$ , we must empty  $S$  to terminate. The choice of  $t_i = T_i$  gives us the desired upper bound for Theorem 5.5,  $w(W_i) < T_i + w_{\max}$ . The proof of Theorem 5.5 is complete.

Finally, we need to redefine the terminology from the proof of theorem of Theorem 1.4 to account for the weights and directedness. Let  $W_1, W_2, \dots, W_k$  be such that  $v_i \in W_i$  for each  $i$ ,  $W_1$  is underweight and no  $W_i$  is overweight.

Define  $S := V(G) - W_1 - W_2 - \dots - W_k$ . For a vertex  $v \in W_i$  we define the **reservoir** of  $v$ , denoted by  $R(v)$ , to be the set of all vertices in  $W_i$  which are connected to  $v_i$  by a directed path in  $G[W_i - \{v\}]$ . Note that  $v \notin R(v)$  and also  $R(v_i) = \emptyset$ . By a **cascade** in  $W_i$  we mean a (possibly null) sequence  $u_1, u_2, \dots, u_m$  of distinct vertices in  $W_i \setminus v_i$  such that  $u_{j+1} \notin R(u_j)$  for  $j = 1, \dots, m-1$ . So for each  $j = 1, \dots, m-1$ , all directed paths from  $u_{j+1}$  to  $u_{j-1}$  in  $G[W_i]$  contain  $u_j$ , where  $u_0$  means  $v_i$ . By a **configuration** we mean a choice of vertex sets  $W_1, W_2, \dots, W_k$  such that  $v_i \in W_i$  for each  $i$ ,  $W_1$  is underweight and no  $W_i$  is overweight, and an assignment of exactly one cascade to each  $W_i$  for  $i = 2, 3, \dots, k$ . By a **cascade vertex** we mean a vertex belonging to one of the cascades in the configuration. For a cascade vertex  $u$ , we define its **proper reservoir** to be  $R'(u) = R(u) - R(v)$ , where  $v$  is the vertex before  $u$  in its cascade. If  $u$  is the first vertex in its cascade, then  $R'(u) = R(u)$ .

Note that each vertex in  $W_2 \cup \dots \cup W_k$  either belongs to a unique proper reservoir, or is not in any reservoir. We define the **rank** of some cascade vertices recursively as follows. Let  $u \in W_i$  be a cascade vertex. If there is an edge from  $u$  to  $W_1$  in  $G$ , then we define the rank of  $u$  to be 1. Otherwise, its rank is the least integer  $r \geq 2$  such that there is a cascade vertex  $u' \in W_j$ , for some  $j \in \{2, 3, \dots, k\} - \{i\}$ , and an edge from  $u$  to  $R(u')$  in  $G$ , where  $u'$  has rank  $r - 1$ . If there is no such edge, then the rank of  $u$  is undefined. The rank of a reservoir or proper reservoir will mean the rank of its corresponding cascade vertex. For an integer  $r \geq 1$ , let  $\rho_r$  denote the total weight of vertices belonging to rank  $r$  reservoirs. A configuration is **valid** if each cascade vertex has well-defined rank and this rank is strictly increasing within a cascade. That is, for each cascade  $u_1, u_2, \dots, u_m$  and integers  $1 \leq i < j \leq m$  the rank of  $u_i$  is strictly smaller than the rank of  $u_j$ . Note that a valid configuration exists trivially by taking each cascade to be the null sequence. For an integer  $r \geq 1$  a valid configuration is  **$r$ -optimal** if, among all valid configurations, it maximizes  $\rho_1$ , subject to that it maximizes  $\rho_2$ , and so on, up to maximizing  $\rho_r$ . If a valid configuration is  $r$ -optimal for all  $r \geq 1$ , we simply say it is **optimal**. Note that an optimal configuration exists since a valid configuration exists. Finally, a **bridge** is an edge with tail in  $S$  and head in a reservoir. The rank of a bridge is the rank of the unique proper reservoir containing the head of the bridge.

These concepts are illustrated in Figures 5.2 and 5.3.

### 5.2.2 Proof of Theorem 5.6

We are ready to begin the proof. Choose an optimal configuration. We will define two disjoint regions in the graph. The *reservoir region* contains  $W_1$  and any vertex belonging to a reservoir. The *non-reservoir region* contains vertices not in  $W_1$ , not equal to a  $v_i$  or a cascade vertex, and not in any reservoir. These regions are illustrated in Figure 5.4. Since  $W_1$  and  $S$  are non-empty, both regions are non-empty. There are exactly  $k - 1$  vertices not in a region: the last vertex in each cascade (or  $v_i$  if  $W_i$  has a null cascade). Note that none





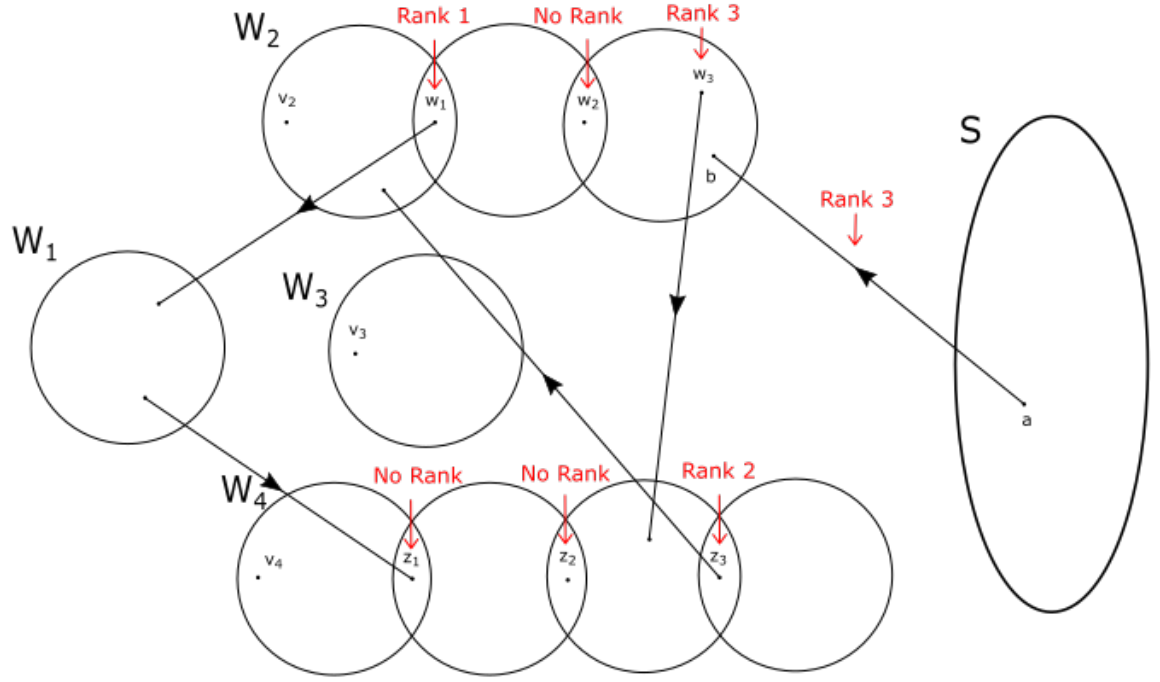


Figure 5.3: Examples of rank. The edge  $ab$  is a bridge. Since  $b \in R'(w_3)$ , the rank of the bridge is the rank of  $w_3$ .

the tail is in  $W_i$  for some  $i = 2, \dots, k$ .

*Proof.* Suppose there is such an edge, and say the tail is  $y \in W_i$  and the head is  $z \in W_j$ , where  $i = 2, \dots, k$  and  $j = 1, 2, \dots, k$ . Then  $z$  is either in  $W_1$  or in the proper reservoir  $R'(u)$  of some cascade vertex  $u$  with defined rank. We create a new configuration by adding the vertex  $y$  to the end of the cascade of  $W_i$  (or create a cascade containing only  $y$ , if it was null before). Now  $y$  has a defined rank via the neighbor  $u$ , say  $y$  is rank  $r$ . Remove all cascade vertices of rank greater than  $r$  from all cascades to maintain the validity of the configuration. Even if this destroys reservoirs of rank greater than  $r$ , we have created a new rank  $r$  reservoir, so we have contradicted the optimality of the original configuration.  $\square$

**Claim 12.** *If there is a bridge, then the conclusion of Theorem 5.6 holds.*

*Proof.* Suppose our optimal configuration contains a bridge. Consider the set of  $r$ -optimal configurations containing bridges of rank  $r$ . Our original configuration shows that this set



is overweight, we will eventually reach a good vertex set. Choose a new configuration with this set replacing  $W_i$ , and remove all cascade vertices of rank greater than  $r$  from all cascades to maintain validity. This may destroy reservoirs of rank greater than  $r$ , but the rank  $r$  reservoir  $R(u)$  gained a new vertex  $s$ , which contradicts  $r$ -optimality. We continue under the assumption that  $W'_i$  is not underweight.

We now know that  $W'_i$  is not underweight and we observe that  $W'_i - \{s\} \subset W_i$  is not overweight. If neither  $W'_i$  nor  $W'_i - \{s\}$  were good, we would have an overweight and underweight set differing by a single vertex  $s$ , contradicting the definition of  $w_{\max}$ . We conclude that at least one of the sets must be good.

Choose a new configuration where either  $W'_i$  or  $W'_i - \{s\}$  (whichever is good, choosing arbitrarily if both are good) replaces  $W_i$ . Remove all cascade vertices of rank greater than  $r - 1$  from all cascades to maintain validity. This new configuration lost at least one reservoir, but all destroyed reservoirs had rank greater than  $r - 1$ , so it is still  $(r - 1)$ -optimal. Further, since  $u$  had rank  $r$  in the previous configuration, there is an edge from  $u$  to a rank  $r - 1$  reservoir (or to  $W_1$ , if  $r = 1$ ). The vertex  $u$  is in  $S$  in the new configuration. If  $r = 1$  this is an edge from  $S$  to  $W_1$  and we can grow  $W_1$  as in Claim 10. If  $r > 1$ , this edge is a rank  $r - 1$  bridge, which contradicts the minimality of  $r$ .  $\square$

The claims cover every possibility for an edge from the non-reservoir region to the reservoir region, so the proof of Theorem 5.6 is complete.  $\square$

## 5.3 Fast Computation of Györi-Lovász Partitions

### 5.3.1 Introduction

The proofs in the previous sections are not known to be polynomial time in general. For  $k = 2$  there is only one cascade and therefore only a linear number of rank combinations (everything has rank 1 or undefined rank), but for higher  $k$  there are an exponential number of valid rank counts.

Here, we provide an alternate approach to the Györi-Lovász theorem, using the ear decomposition and planar chain decomposition. This allows us to construct the Györi-Lovász partitions in polynomial time for  $k = 2, 3, 4$ .

### 5.3.2 Proof for $k = 2$

Note that the original proof of the Györi-Lovász theorem is already polynomial time when  $k = 2$ , but we provide an alternate approach here, as it will be instructional for further cases.

Recall the open ear decomposition.

**Definition 5.7.** For a graph  $G$ , an *open ear decomposition* rooted at  $r$  is a sequence of subgraphs  $G_1, G_2, \dots, G_m$ , called *ears*, such that:

- $E(G_1), E(G_2), \dots, E(G_m)$  partition  $E(G)$ .
- $G_1$  is a cycle containing  $r$ .
- For  $i = 2, \dots, m$ ,  $G_i$  is a path whose intersection with  $G_1 \cup \dots \cup G_{i-1}$  is exactly its two ends.

It is well known that a graph  $G$  is 2-connected if and only if it has an open ear decomposition, and if so, one can be constructed at any root  $r$ . Itai and Rodeh use this decomposition to order the vertices as follows.

**Theorem 5.8** ([13]). *Suppose  $G_1, G_2, \dots, G_m$  is an open ear decomposition rooted at  $r$ . Then there is an ordering of the vertices of  $G$  so that  $r$  is the least vertex, some neighbor of  $r$  in  $G_1$  is the greatest vertex, and every other vertex is adjacent to both a lesser vertex and a greater vertex. Further, the lesser and greater neighbors can be chosen to be in either the same ear or an earlier ear.*

*Proof of Theorem 1.4 when  $k = 2$ .* We can use the ordering to easily produce a Györi-Lovász partition. Suppose we have a 2-connected graph  $G$  and  $v_1, v_2, n_1, n_2$  as in Theorem 1.4.

Create a graph  $G'$  by adding the edge  $v_1v_2$ , if it does not already exist. Apply Theorem 5.8 to obtain an ordering of the vertices with  $v_1$  as the least vertex and  $v_2$  as the greatest vertex.

Assign the greatest  $n_2$  vertices to  $W_2$ , and the remaining  $n_1$  vertices to  $W_1$ . The properties of the ordering ensure that for any vertex in  $W_2$ , we can repeatedly move to greater neighbors (which will also be in  $W_2$ ) and eventually arrive at  $v_2$ . So,  $W_2$  is connected to  $v_2$  and therefore connected. Similarly,  $W_1$  is connected to  $v_1$  and therefore connected. We have the desired partition.  $\square$

**Theorem 5.9.** *There exists an algorithm with the following specifications:*

**Input:** A 2-connected graph  $G$  on  $n$  vertices and  $m$  edges, two distinct vertices  $v_1, v_2$ , and natural numbers  $n_1, n_2$  such that  $n_1 + n_2 = n$ .

**Output:** A partition of  $V(G)$  into  $V_1, V_2$ , such that, for  $i = 1, 2$ ,  $G[V_i]$  is connected,  $v_i \in V_i$  and  $|V_i| = n_i$ .

**Running time:**  $O(m)$

*Proof.* Create a graph  $G'$  by adding the edge  $v_1v_2$ , if it does not already exist. Construct an  $s - t$  numbering, with  $v_1$  as the low vertex and  $v_2$  as the high vertex, using the  $O(m)$  algorithm of Even and Tarjan [8]. Assign the smallest  $n_1$  vertices in the list to  $V_1$ , and the remaining  $n_2$  vertices to  $V_2$ .  $\square$

### 5.3.3 Proof for $k = 3$

We can do something similar for  $k = 3$ . We need to define a nonseparating open ear decomposition as follows.

**Definition 5.10.** An open ear decomposition  $G_1, G_2, \dots, G_m$  rooted at  $r$  is *nonseparating* if

- $G_m$  is a path of length one containing  $r$ ,
- for  $i = 1, \dots, m - 1$ , every internal vertex of  $G_i$  is in some  $G_j$ , with  $j > i$ , AND

- for  $i = 1, \dots, m - 1$ , if  $G_i$  has length 1, then at least one endpoint of  $G_i$  is in some  $G_j$ , with  $j > i$ .

Cheriyān and Maheshwari [4] show that a graph is 3-connected if and only if it has such a decomposition, and if so, one can be constructed from any root  $r$ .

*Proof of Theorem 1.4 when  $k = 3$ .* Suppose we have a 3-connected graph  $G$  and  $v_1, v_2, v_3, n_1, n_2, n_3$  as in Theorem 1.4.

Create a graph  $G'$  by adding a new vertex  $r$  whose neighbors are exactly  $v_1, v_2$ , and  $v_3$  (note that  $G'$  is 3-connected). Construct a nonseparating open ear decomposition rooted at  $r$ , then apply Theorem 5.8 to obtain an ordering of the vertices with  $r$  as the least vertex. Without loss of generality,  $v_1$  is the greatest vertex and  $v_2$  is the second-least vertex. This leaves  $v_3$  as the endpoint of the final ear  $G_m$ .

Let  $i^*$  be the largest index such that  $|V(G_1 \cup G_2 \cdots \cup G_{i^*-1})| \leq n_1 + n_2 + 1$ , and call the difference  $n^* = (n_1 + n_2 + 1) - |V(G_1 \cup G_2 \cdots \cup G_{i^*-1})|$ .

First, suppose  $n^* = 0$ . Then  $G_1, G_2, \dots, G_{i^*}$  is an open ear decomposition of  $G_1 \cup G_2 \cdots \cup G_{i^*}$  with  $n_1 + n_2 + 1$  vertices, with  $v_1$  and  $v_2$  as the greatest and second-least vertices, respectively. So, we can construct  $W_1$  and  $W_2$  exactly as we did in the case  $k = 2$ . The remaining  $n_3$  vertices of  $G'$  will form  $W_3$ . Recall that the original ear decomposition is nonseparating. So, from any vertex in  $W_3$ , we can repeatedly choose a neighbor in  $G'$  which belongs to a later ear, until we reach  $v_3$ . So,  $W_3$  is connected to  $v_3$  and therefore connected.

So we can assume  $n^* > 0$ . Build an open ear decomposition rooted at  $r$  on  $G_1 \cup G_2 \cdots \cup G_{i^*-1}$ , then extend it with a new ear  $G^*$ , having  $n^*$  internal vertices and the same ends as  $G_{i^*}$ . This graph has  $n_1 + n_2 + 1$  vertices by definition, so we can construct  $W_1$  and  $W_2$  as we did in the case  $k = 2$ .

We need to map the vertices of  $G^*$  to vertices of  $G_{i^*}$ . Note that, since  $G^*$  is the last ear of the truncated decomposition, every vertex in  $(G_1 \cup G_2 \cdots \cup G_{i^*-1}) \cap W_1$  has a path to  $v_1$  which does not pass through the interior of  $G^*$ , and similarly for  $W_2$ . Thus, we will not

disconnect  $(G_1 \cup G_2 \cdots \cup G_{i^*-1}) \cap W_1$  or  $(G_1 \cup G_2 \cdots \cup G_{i^*-1}) \cap W_2$  by mapping  $G^*$  to  $G_{i^*}$ .

We map the vertices as follows. If  $G^* \cap W_1$  is nonempty, then choose an end of  $G^*$  in  $W_1$  and map the internal vertices of  $G^* \cap W_1$  to internal vertices of  $G_{i^*}$  adjacent to that end. Do the same for  $W_2$ . The remaining vertices of  $G_{i^*}$ , together with all vertices in later ears, form  $W_3$ . Again, the fact that the original ear decomposition is nonseparating implies  $W_3$  is connected to  $v_3$ . Carrying these assignments back to the original graph  $G$ , we have the desired partition.  $\square$

**Theorem 5.11.** *There exists an algorithm with the following specifications:*

**Input:** *A 3-connected graph  $G$  on  $n$  vertices and  $m$  edges, three distinct vertices  $v_1, v_2, v_3$ , and natural numbers  $n_1, n_2, n_3$  such that  $n_1 + n_2 + n_3 = n$ .*

**Output:** *A partition of  $V(G)$  into  $V_1, V_2, V_3$ , such that, for  $i = 1, 2, 3$ ,  $G[V_i]$  is connected,  $v_i \in V_i$  and  $|V_i| = n_i$ .*

**Running time:**  $O(m)$

*Proof.* Create a graph  $G'$  by adding a new vertex  $r$  whose neighbors are exactly  $v_1, v_2, v_3$ . Construct a nonseparating ear decomposition rooted at  $r$  and use it to number the vertices as in [18], which takes  $O(m)$  time. Permute the  $v_i$ 's (and corresponding  $n_i$ 's) so that  $v_1$  is the greatest vertex,  $v_2$  is the second-least vertex, and  $v_3$  is the endpoint of the final ear.

Loop through ears until finding the maximum index  $i^*$  such that  $|V(G_1 \cup G_2 \cdots \cup G_{i^*-1})| \leq n_1 + n_2 + 1$ , and call the difference  $n^* = (n_1 + n_2 + 1) - |V(G_1 \cup G_2 \cdots \cup G_{i^*-1})|$ . This takes  $O(m)$  time.

If  $n^* = 0$ , apply Theorem 5.9 to  $G_1 \cup G_2 \cdots \cup G_{i^*}$  to construct  $W_1$  and  $W_2$  in  $O(m)$  time, and call the remainder of the graph  $W_3$ .

If  $n^* > 0$ , create a new ear  $G^*$ , having  $n^*$  internal vertices and the same ends as  $G_{i^*}$ . Apply Theorem 5.9 to  $G_1 \cup G_2 \cdots \cup G_{i^*-1} \cup G^*$  to construct a tentative  $W_1^*$  and  $W_2^*$  in  $O(m)$  time.

If  $G^* \cap W_1$  is nonempty, then choose an end of  $G^*$  in  $W_1^*$  and map the internal vertices of  $G^* \cap W_1$  to internal vertices of  $G_{i^*}$  adjacent to that end. Do the same for  $W_2$ . The remaining vertices of  $G_{i^*}$ , together with all vertices in later ears, form  $W_3$ .  $\square$

**Corollary 5.12.** *Suppose we have a graph  $G$  on  $n$  vertices and  $m$  edges,  $v_1, v_2, v_3 \in V(G)$ , and natural numbers  $n_1, n_2, n_3$  so that  $n_1 + n_2 + n_3 = n$ . Assume  $G$  is 2-connected and is 3-connected to the vertex set  $\{v_1, v_2, v_3\}$ . Then there exists a partition of  $V(G)$  into  $V_1, V_2, V_3$ , such that, for  $i = 1, 2, 3$ ,  $G[V_i]$  is connected,  $v_i \in V_i$  and  $|V_i| = n_i$ . Further, such a partition can be found in  $O(m)$  time.*

*Proof.* Create a graph  $G'$  from  $G$  by adding an edge between every pair of vertices in  $\{v_1, v_2, v_3\}$ . Note that  $G'$  is 3-connected, since any 2-cut in  $G$  has members of  $\{v_1, v_2, v_3\}$  on both sides. Apply Theorem 5.11 to  $G'$ , using the same partition for  $G$ . The lack of edges between  $\{v_1, v_2, v_3\}$  in  $G$  does not affect connectivity because each of  $\{v_1, v_2, v_3\}$  is in a different part.  $\square$

#### 5.3.4 Proof for $k = 4$

We can extend this technique one step further to  $k = 4$ , using the planar chain decomposition of Curran, Lee, and Yu [6, 5, 7]. We reproduce the relevant definitions, figure, and theorem here.

**Definition 5.13** ([6]). A connected graph  $H$  is a *chain* if for some integer  $k \geq 2$ , there exist subgraphs  $B_1, \dots, B_k$  of  $H$  and vertices  $b_0, b_1, \dots, b_k$  of  $H$  such that:

1. for  $1 \leq i \leq k$ ,  $B_i$  is 2-connected or  $B_i$  is induced by an edge of  $H$ ,
2.  $V(H) = \bigcup_{i=1}^k V(B_i)$  and  $E(H) = \bigcup_{i=1}^k E(B_i)$ ,
3.  $V(B_i) \cap V(B_{i+1}) = \{b_i\}$  for  $1 \leq i \leq k-1$ , and  $V(B_i) \cap V(B_j) = \emptyset$  for  $1 \leq i < i+2 \leq j \leq k$ .

We use the notation  $H := b_0 B_1 b_1 \dots b_{k-1} B_k b_k$ . Each subgraph  $B_i$  is called a *piece* of  $H$ .



**Definition 5.14** ([6, Definition 1.4]). Let  $G$  be a graph and let  $H := b_0 B_1 b_1 \dots b_{k-1} B_k v_k$  be a chain. If  $H$  is an induced subgraph of  $G$ , then we say that  $H$  is a *chain* in  $G$ . We say that  $H$  is a *planar chain* in  $G$  if, for each  $1 \leq i \leq k$  with  $|V(B_i)| \geq 3$  (or equivalently,  $B_i$  is 2-connected), there exist distinct vertices  $x_i, y_i \in V(G) - V(H)$  such that  $(G[V(B_i) \cup \{x_i, y_i\}] - x_i y_i, x_i, b_{i-1}, y_i, b_i)$  is planar, and  $B_i - \{b_{i-1}, b_i\}$  is a component of  $G - \{x_i, y_i, b_{i-1}, b_i\}$ . We also say that  $H$  is a *planar  $b_0$ - $b_k$  chain*.

**Definition 5.15** ([6, Definition 4.2]). A connected graph  $H$  is a *cyclic chain* if for some integer  $k \geq 2$ , there exist subgraphs  $B_1, \dots, B_k$  of  $H$  and vertices  $b_1, \dots, b_k$  of  $H$  such that:

1. for  $1 \leq i \leq k$ ,  $B_i$  is 2-connected or  $B_i$  is induced by an edge of  $H$ ,
2.  $V(H) = \bigcup_{i=1}^k V(B_i)$  and  $E(H) = \bigcup_{i=1}^k E(B_i)$ ,
3. if  $k = 2$ , then  $V(B_1) \cap V(B_2) = \{b_1, b_2\}$  and  $E(B_1) \cap E(B_2) = \emptyset$ , and
4. if  $k \geq 3$ , then  $V(B_i) \cap V(B_{i+1}) = \{b_i\}$  for  $1 \leq i \leq k$ , where  $B_{k+1} := B_1$ , and  $V(B_i) \cap V(B_j) = \emptyset$  for  $1 \leq i < i+2 \leq j \leq k$  and  $(i, j) \neq (1, k)$ .

We usually fix one of the vertices  $b_1, \dots, b_k$  as the root of  $H$ , say,  $b_k$ , and we use the notation  $H := b_0 B_1 b_1 \dots b_{k-1} B_k b_k$  to indicate that  $H$  is a cyclic chain rooted at  $b_0 (= b_k)$ . Each subgraph  $B_i$  is called a *piece* of  $H$ .

**Definition 5.16** ([6, Definition 4.3]). Let  $G$  be a graph and let  $H := b_0 B_1 b_1 \dots b_{k-1} B_k b_k$  be a cyclic chain rooted at  $b_0 = b_k$ . If  $H$  is an induced subgraph of  $G$ , then we say that  $H$  is a *cyclic chain* in  $G$ . We say that  $H$  is a *planar cyclic chain* in  $G$  if for each  $1 \leq i \leq k$  with  $|V(B_i)| \geq 3$  (or equivalently,  $B_i$  is 2-connected), there exist distinct vertices  $x_i, y_i \in V(G) - V(H)$  such that  $(G[V(B_i) \cup \{x_i, y_i\}] - x_i y_i, x_i, b_{i-1}, y_i, b_i)$  is planar, and  $B_i - \{b_{i-1}, b_i\}$  is a component of  $G - \{x_i, y_i, b_{i-1}, b_i\}$ .

**Definition 5.17** ([5, Definition 1.1]). Let  $G$  be a graph, let  $F$  be a subgraph of  $G$ , and let  $r \in V(F)$ . Let  $H$  be a planar  $x$ - $y$  chain in  $G$  such that  $V(H) - \{x, y\} \subseteq V(G) - V(F)$ .

1.  $H$  is an *up  $F$ -chain* if  $\{x, y\} \subseteq V(F)$  and  $N_G(H - \{x, y\}) \subseteq (V(G) - V(F - r)) \cup \{x, y\}$ ;
2.  $H$  is a *down  $F$ -chain* if  $\{x, y\} \subseteq V(G) - V(F - r)$  and  $N_G(H - \{x, y\}) \subseteq V(F - r) \cup \{x, y\}$ ; and
3.  $H$  is an *elementary  $F$ -chain* if  $\{x, y\} \subseteq V(F)$  and  $H$  is an  $x$ - $y$  path of length two.

In any of the three cases above we say that  $H$  is a *planar  $x$ - $y$   $F$ -chain in  $G$*  (or simply, a *planar  $F$ -chain*). For an  $x$ - $y$  chain  $H$  we let  $I(H) := V(H) - \{x, y\}$ , and for a cyclic chain  $H$  we let  $I(H) := V(H)$ .

**Definition 5.18** ([5, Definition 1.2]). Let  $G$  be a graph, let  $F$  be a subgraph of  $G$ , and let  $r \in V(F)$ . Suppose that  $\{w_1, w_2, w_3\} \subseteq V(G) - V(F)$  induces a triangle  $T$  in  $G$ , and for each  $1 \leq i \leq 3$ ,  $w_i$  has exactly one neighbor  $x_i$  in  $V(F - r)$  and exactly one neighbor  $y_i$  in  $V(G) - (V(F) \cup V(T))$  (thus, each  $w_i$  has degree four in  $G$ ). Moreover, assume that  $x_1, x_2, x_3$  are distinct and  $y_1, y_2, y_3$  are distinct. Then we say that  $H := T + x_1, x_2, x_3, w_1x_1, w_2x_2, w_3x_3$  is a *triangle  $F$ -chain* in  $G$ . We let  $I(H) := \{w_1, w_2, w_3\}$ .

**Definition 5.19** ([5, Definition 1.3]). Let  $G$  be a graph, let  $F$  be a subgraph of  $G$ , and let  $r \in V(F)$ . By a *good  $F$ -chain in  $G$* , we mean an up  $F$ -chain, a down  $F$ -chain, an elementary  $F$ -chain, or a triangle  $F$ -chain.

**Definition 5.20** ([5, Definition 1.4]). Let  $G$  be a graph, let  $r \in V(G)$ , and let  $H_1, \dots, H_t$  be chains in  $G$ , where  $t \geq 2$ . We say that  $(H_1, \dots, H_t)$  is a *nonseparating chain decomposition of  $G$  rooted at  $r$*  if the following conditions hold:

1.  $H_1$  is a planar cyclic chain in  $G$  rooted at  $r$ ;
2. for each  $i = 2, \dots, t - 1$ ,  $H_i$  is a good  $G \left[ \bigcup_{j=1}^{i-1} I(H_j) \right]$ -chain in  $G$ ;
3.  $H_t = G - \left( \bigcup_{j=1}^{t-1} I(H_j) - \{r\} \right)$  is a planar cyclic chain in  $G$  rooted at  $r$ ; and

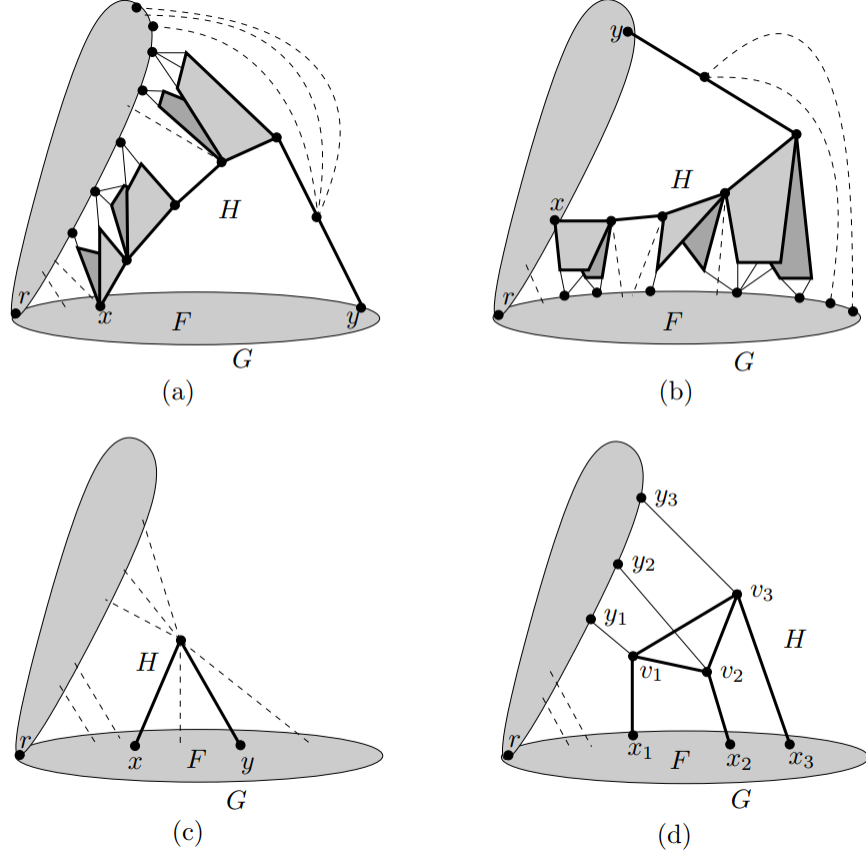


Figure 5.5: (Figure 4 of [7]). An illustration of (a) an up chain, (b) a down chain, (c) an elementary chain, and (d) a triangle chain.

4. for each  $i = 1, \dots, t - 1$ , both  $G \left[ \bigcup_{j=1}^i I(H_j) \right]$  and  $G - \left( \bigcup_{j=1}^i I(H_j) - \{r\} \right)$  are 2-connected.

The chains  $H_2, \dots, H_{t-1}$  are called *internal chains* of the nonseparating chain decomposition. If  $ra$  is a piece of  $H_1$ , then we say that  $H_1, \dots, H_t$  is a *nonseparating chain decomposition* of  $G$  starting at  $ra$ .

**Theorem 5.21** ([5, Theorem 1.5]). *Let  $G$  be a 4-connected graph, let  $r \in V(G)$ , and let  $ra \in E(G)$ . Then  $G$  has a nonseparating chain decomposition rooted at  $r$  starting at  $ra$ , and such a decomposition can be found in  $O(|V(G)|^2 |E(G)|)$  time.*

Next, we prove some results about the symmetry of the planar chain decomposition.

**Lemma 5.22.** Suppose  $(H_1, \dots, H_t)$  is a nonseparating chain decomposition of  $G$  rooted at  $r$ . Then  $G - \left(\bigcup_{j=1}^i I(H_j) - \{r\}\right) = G \left[\bigcup_{j=i+1}^t I(H_j)\right]$  for each  $i = 1, 2, \dots, t-1$ .

*Proof.* Both graphs are induced subgraphs of  $G$ , so it suffices to show that they have the same vertex set, i.e. that  $\bigcup_{j=1}^i I(H_j) - \{r\}$  and  $\bigcup_{j=i+1}^t I(H_j)$  partition  $V(G)$ . It follows from condition 3 in Definition 5.20 that the sets cover  $V(G)$ , so it remains to show they are disjoint.

Assume for the sake of contradiction that  $v \in \left(\bigcup_{j=1}^i I(H_j) - \{r\}\right) \cap \left(\bigcup_{j=i+1}^t I(H_j)\right)$ . Then  $v \neq r$  and  $v \in I(H_{j_1}) \cap I(H_{j_2})$  for some  $j_1 < j_2$ . By Definition 5.20,  $H_{j_2}$  is a good  $G \left[\bigcup_{j=1}^{j_2-1} I(H_j)\right]$ -chain. But then by Definition 5.19,

$$I(H_{j_2}) \subseteq V(G) - \bigcup_{j=1}^{j_2-1} I(H_j) \subseteq V(G) - I(H_{j_1})$$

This contradicts the fact that  $v \in I(H_{j_1}) \cap I(H_{j_2})$ . The proof is complete.  $\square$

**Corollary 5.23.** Suppose  $(H_1, \dots, H_t)$  is a nonseparating chain decomposition of  $G$  rooted at  $r$ , and  $H_i$  is an elementary  $\left[\bigcup_{j=1}^{i-1} I(H_j)\right]$ -chain. Then  $H_i$  is an elementary  $G \left[\bigcup_{j=i+1}^t I(H_j)\right]$ -chain.

*Proof.* Let  $\{v\} = I(H_i)$ . By Lemma 5.22,  $G - \left(\bigcup_{j=1}^i I(H_j) - \{r\}\right) = G \left[\bigcup_{j=i+1}^t I(H_j)\right]$  so we need only prove that  $v$  has degree at least two in  $G - \left(\bigcup_{j=1}^i I(H_j) - \{r\}\right)$ . But by Definition 5.20,  $G - \left(\bigcup_{j=1}^{i-1} I(H_j) - \{r\}\right)$  is 2-connected so  $v$  has degree 2 in this subgraph. Since  $I(H_i) = \{v\}$ , both neighbors must be in  $G - \left(\bigcup_{j=1}^i I(H_j) - \{r\}\right)$ .  $\square$

**Corollary 5.24.** Suppose  $(H_1, \dots, H_t)$  is a nonseparating chain decomposition of  $G$  rooted at  $r$ . Then  $(H_t, \dots, H_1)$  is a nonseparating chain decomposition of  $G$  rooted at  $r$ .

*Proof.* We consider the four conditions for a nonseparating chain decomposition in Definition 5.20.

1.  $H_t$  is a planar cyclic chain in  $G$  rooted at  $r$  by the definition of  $H_t$ .

2. We can reclassify up chains and down chains, down chains as up chains, and triangle chains as triangle chains immediately from Lemma 5.22. For elementary chains, we may reclassify them as elementary chains using Corollary 5.23.
3.  $H_1$  is a planar cyclic chain of  $G$  rooted at  $r$  by the definition of  $H_1$ , and  $H_1 = G - \left( \bigcup_{j=2}^t I(H_j) - \{r\} \right)$  by Lemma 5.22 with  $i = 1$ .
4. Each  $G \left[ \bigcup_{j=i}^t I(H_j) \right]$  and  $G - \left( \bigcup_{j=i}^t I(H_j) - \{r\} \right)$  are 2-connected by Lemma 5.22.

□

We are ready to prove the main result.

*Proof of Theorem 1.4 when  $k = 4$ .* Suppose we have a graph  $G$  on  $n$  vertices,  $v_1, v_2, v_3, v_4 \in V(G)$ , and natural numbers  $n_1, n_2, n_3, n_4$  so that  $n_1 + n_2 + n_3 + n_4 = n$ . Assume  $G$  is 2-connected and is 4-connected to the vertex set  $\{v_1, v_2, v_3, v_4\}$ . We proceed by induction on  $n$ . In the base case we have only one possible partition,  $W_i = \{v_i\}$  for  $i = 1, 2, 3, 4$ .

Create a graph  $G'$  by adding a  $K_4$  between  $v_1, v_2, v_3, v_4$ , as well as a new vertex  $r$  whose neighbors are exactly  $v_1, v_2, v_3$ , and  $v_4$  (note that  $G'$  is 4-connected). Construct a nonseparating chain decomposition  $H_1, H_2, \dots, H_m$  of  $G'$  rooted at  $r$ . Without loss of generality,  $v_1$  and  $v_2$  are in the first chain and  $v_3$  and  $v_4$  are in the last chain.

Let  $i^*$  be the largest index such that  $\left| G \left[ \bigcup_{j=1}^{i^*-1} I(H_j) \right] \right| \leq n_1 + n_2 + 1$ , and call the difference  $n^* = (n_1 + n_2 + 1) - \left| G \left[ \bigcup_{j=1}^{i^*-1} I(H_j) \right] \right|$ .

First, suppose  $n^* = 0$ . Then  $G \left[ \bigcup_{j=1}^{i^*-1} I(H_j) \right]$  is a 2-connected graph with  $n_1 + n_2 + 1$  vertices containing  $v_1$  and  $v_2$ . So, we can build an open ear decomposition rooted at  $r$  and construct  $W_1$  and  $W_2$  as we did in the case  $k = 2$ . Similarly,  $G - \left( \bigcup_{j=1}^{i^*} I(H_j) - \{r\} \right)$  is a 2-connected graph with  $n_3 + n_4 + 1$  vertices containing  $v_3$  and  $v_4$ . So, we can build an open ear decomposition rooted at  $r$  and construct  $W_3$  and  $W_4$  as we did in the case  $k = 2$ . We have the desired partition.

So we can assume  $n^* > 0$ . Then  $H_{i^*}$  is not an elementary chain. By Corollary 5.24,  $H_t, H_{t-1}, \dots, H_1$  is also a chain decomposition. So, by reversing the decomposition if

necessary, we may assume that  $H_{i^*}$  is either a triangle chain or an up chain, and that  $n^* = 1$  in the former case.

Suppose  $H_{i^*}$  is a triangle chain with vertices  $x_1, x_2, x_3, w_1, w_2, w_3$  and neighbors  $y_1, y_2, y_3$ , and  $n^* = 1$ . Build an open ear decomposition rooted at  $r$  on  $G \left[ \bigcup_{j=1}^{i^*-1} I(H_j) \right]$ , then extend it with a new ear  $G^*$ , having 1 internal vertex and ends chosen from  $x_1, x_2, x_3$  arbitrarily. The extended ear decomposition has  $n_1 + n_2 + 1$  vertices by definition, so we can construct  $W_1$  and  $W_2$  as we did in the case  $k = 2$ . Without loss of generality, the internal vertex of  $G^*$  is assigned to  $W_1$ , which also contains  $x_1$ . We then map the internal vertex to  $w_1$  in the original graph, so that  $W_1$  remains connected.

We do something similar with the remaining vertices. Build an open ear decomposition rooted at  $r$  on  $G - \left( \bigcup_{j=1}^{i^*} I(H_j) - \{r\} \right)$ , then extend it with a new ear  $G^{**}$ , consisting of vertices  $w_2, w_3$  and ends  $y_2, y_3$ . The extended ear decomposition has  $n_3 + n_4 + 1$  vertices by definition, so we can construct  $W_3$  and  $W_4$  as we did in the case  $k = 2$ . We have the desired partition.

So we can assume  $H_{i^*}$  is an up chain  $b_0 B_1 b_1 \dots b_{k-1} B_k b_k$ . Build an open ear decomposition rooted at  $r$  on  $G \left[ \bigcup_{j=1}^{i^*-1} I(H_j) \right]$ , then extend it with a new ear  $G^*$ , having  $n^*$  internal vertices and ends  $b_0$  and  $b_k$ . The extended ear decomposition has  $n_1 + n_2 + 1$  vertices by definition, so we can construct a tentative  $W_1^*$  and  $W_2^*$  as we did in the case  $k = 2$ .

We introduce some notation for  $H_{i^*}$ , organizing its pieces in the same way that we organized the chains above. Suppose we have assigned  $p_1$  internal vertices of  $H^*$  to  $W_1^*$ , and the remaining  $p_2$  to  $W_2^*$ . Without loss of generality,  $p_1 > 0$  so there is an end of  $H^*$  in  $W_1$ , say  $b_0$ .

Let  $q_1$  be the largest index so that  $|V(B_1 \cup B_2 \cup \dots \cup B_{q_1-1}) - \{b_0, b_\ell\}| \leq p_1$ , and call the difference  $n_1^* = p_1 - |V(B_1 \cup B_2 \cup \dots \cup B_{q_1-1}) - \{b_0, b_\ell\}|$ . Similarly, let  $q_2$  be the minimal index so that  $|V(B_{q_2+1} \cup B_{q_2+2} \cup \dots \cup B_\ell) - \{b_0, b_\ell\}| \leq p_2$ , and call the difference  $n_2^* = p_2 - |V(B_{q_2+1} \cup B_{q_2+2} \cup \dots \cup B_\ell) - \{b_0, b_\ell\}|$ . Then  $q_1 \leq q_2$ .

We have two tentative parts  $W_1^*, W_2^*$  which are connected, contain the corresponding

$v_i$ , and have the correct size, but are not subgraphs of  $G'$ ; they use the internal vertices of  $G^*$  in place of the chain  $H_{i^*}$ . We will resolve this by creating final parts  $W_1, W_2$ . For vertices outside of  $H_{i^*}$ , we simply assign every vertex from  $W_j^* - G^*$  to  $W_j$  for  $j = 1, 2$ . Note that, since  $G^*$  is the last ear of its ear decomposition, each  $W_j$  is connected at this stage despite lacking vertices from  $G^*$ .

We further assign  $\left(\bigcup_{j=1}^{q_1-1} V(B_j)\right) - \{b_\ell\}$  to  $W_1$  and  $\left(\bigcup_{j=q_2+1}^\ell V(B_j)\right) - \{b_0\}$  to  $W_2$ . Now  $W_1$  requires  $n_1^*$  more vertices, which we will obtain (in a process described later) from  $B_{q_1}$  and connect via  $b_{q_1-1}$ , and similarly  $W_2$  requires  $n_2^*$  more vertices, which we will obtain from  $B_{q_2}$  and connect via  $b_{q_2}$ .

Before we can choose the final vertices for  $W_1$  and  $W_2$ , we need to discuss  $W_3$  and  $W_4$ . Build an open ear decomposition rooted at  $r$  on  $G - \left(\bigcup_{j=1}^{i^*} I(H_j) - \{r\}\right)$ , then extend it with new ears as follows. Let  $I$  be the set of all indices  $i \in \{q_1, q_1 + 1, \dots, q_2\}$  such that either  $B_i$  is 2-connected, or else  $i > q_1$  and  $B_i, B_{i-1}$  are each induced by an edge. For each  $i \in I$ , create an ear  $G_i^{**}$  with the following specifications:

- If  $B_i$  is two-connected, choose the ends of  $G_i^{**}$  to be the vertices in  $\mathcal{N}(B_i) \cap (V(G') - V(F - r))$ . If not, then  $B_{i-1}$  and  $B_i$  are each induced by an edge, so  $b_{i-1}$  has at least two neighbors in  $V(G') - V(F - r)$ . Choose two of these neighbors as the ends of  $G_i^{**}$ .
- Create

$$|B_i| - \mathbb{1}_{i+1 \in I} - (n_1^* + 1) \cdot \mathbb{1}_{i=q_1} - (n_2^* + 1) \cdot \mathbb{1}_{i=q_2}$$

internal vertices in  $G_i^{**}$ . Note that this is the number of vertices in  $B_i$  which will be assigned to  $W_3$  and  $W_4$ , except we exclude  $b_i$  if  $i + 1 \in I$ , since it will be included in the count for  $G_{i+1}^{**}$  (or in the case  $i = q_2$ ,  $b_{q_2}$  has already been assigned to  $W_2$ ). Thus the total number of internal vertices in the  $G_i^{**}$ s equals the total number of vertices in  $B_{q_1} \cup B_{q_1+1} \cup \dots \cup B_{q_2}$  which will be assigned to  $W_3$  and  $W_4$ .

The extended ear decomposition has  $n_3 + n_4 + 1$  vertices by definition, so we can

construct a tentative  $W_3^*$  and  $W_4^*$  as we did in the case  $k = 2$ .

We have two more tentative parts  $W_3^*$ ,  $W_4^*$  which are connected, contain the corresponding  $v_i$ , and have the correct size, but are not subgraphs of  $G'$ ; they use the internal vertices of the  $G_i^{**}$ s in place of the chain  $H_{i^*}$ . We will resolve this by creating final parts  $W_3, W_4$ . For vertices outside of  $H_{i^*}$ , we simply assign every vertex from  $W_j^* - \bigcup_{i \in I} G_i^{**}$  to  $W_j$  for  $j = 3, 4$ . Note that, since the  $G_i^{**}$ s are internally disjoint and were added to the end of an existing ear decomposition, each  $W_j$  is connected at this stage despite lacking vertices from  $\bigcup_{i \in I} G_i^{**}$ .

Suppose  $q_1 = q_2$ . Then the remaining vertices to be distributed are simply the vertices of  $B_{q_1} = B_{q_2}$ , a planar proper subgraph of  $G'$  which is 2-connected and 4-connected to its boundary by the definition of a planar chain. Further, any part still requiring vertices contains one of the boundary vertices. Therefore we may apply the inductive hypothesis, using the boundary vertices of  $B_{q_1} = B_{q_2}$  as  $v_1, v_2, v_3, v_4$  in the inductive step, to distribute  $B_{q_1} = B_{q_2}$  among  $W_1, W_2, W_3, W_4$  while keeping each part connected. We have the desired partition.

So we may assume  $q_1 < q_2$ . We will iterate through the pieces of the unassigned region, first distributing the vertices in  $B_{q_1}$ , then the vertices in  $B_{q_1+1}$ , and so on up to  $B_{q_2}$ . When we reach some  $B_i$  with  $i \notin I$ , there are no vertices to distribute, as  $b_{i-1}$  was distributed with  $B_{i-1}$  (unless  $i = q_1$ , in which case  $b_{q_1-1}$  was already assigned to  $W_1$ ) and  $b_i$  will be distributed with  $B_{i+1}$  (unless  $i = q_2$ , in which case  $b_{q_2+1}$  was already assigned to  $W_2$ ).

- If  $q_1 \in I$ , we distribute the vertices in  $B_{q_1}$  as follows.

- Set  $v_1^{q_1} = b_{q_1-1}$  and  $n_1^{q_1} = n_1^* + 1$ .
- If  $q_1 + 1 \in I$ , set  $v_2^{q_1} = b_{q_1}^{q_1}$  and  $n_2^{q_1} = 1$ .
- Use the ends of  $G_{q_1}^{**}$  as  $v_3^{q_1}$  and  $v_4^{q_1}$ . If all of  $G_{q_1}^{**}$  was assigned to the same tentative part, say  $W_j^*$  for  $j \in \{3, 4\}$ , then set  $n_j^{q_1} = |G_{q_1}^{**} \cap W_j^*| - 1$  and  $n_{7-j}^{q_1} = 1$ . Otherwise, without loss of generality, we choose  $v_3^{q_1}$  to be the end of



$G_{q_1}^{**}$  in  $W_3^*$ , and  $v_4^{q_1}$  to be the end of  $G_{q_1}^{**}$  in  $W_4^*$ . Then we set  $n_3^{q_1} = |G_{q_1}^{**} \cap W_3^*|$  and  $n_4^{q_1} = |G_{q_1}^{**} \cap W_4^*|$ .

Distribute the vertices between  $W_1$ ,  $W_3$ , and  $W_4$  using either Corollary 5.12 (if  $i+1 \notin I$ ) or the inductive step (if  $i+1 \in I$ ). We have finished distributing  $B_i$ .

- For each  $i \in I - \{q_1, q_2\}$ , we distribute  $B_i$  as follows. We first consider whether all of  $G_i^{**}$  was assigned to the same tentative part, say  $W_j^*$  for  $j \in \{3, 4\}$ . If so, then assign every vertex in  $B_i - b_i$  to  $W_j$ . If  $i+1 \notin I$ , further assign  $b_i$  to  $W_j$ . So we may assume that  $G_i^{**}$  contains vertices assigned to both  $W_3^*$  and  $W_4^*$ . We distribute the vertices as follows.

- If  $i+1 \in I$ , set  $v_2^i = b_i$  and  $n_2^i = 1$ .
- Set the end of  $G_i^{**}$  in  $W_3$  as  $v_3^i$ .
- Set the end of  $G_i^{**}$  in  $W_4$  as  $v_4^i$ .

Distribute the vertices between  $W_3$  and  $W_4$  using either Theorem 1.4 with  $k = 2$  (if  $i+1 \notin I$ ) or Corollary 5.12 (if  $i+1 \in I$ ). We have finished distributing  $B_i$ .

- For  $B_{q_2}$ , we distribute vertices as follows.

- Set  $v_2^{q_2} = b_{q_2}$  and  $n_2^{q_2} = n_2^* + 1$ .
- Use the ends of  $G_{q_1}^{**}$  as  $v_3^{q_2}$  and  $v_4^{q_2}$ . If all of  $G_{q_2}^{**}$  was assigned to the same tentative part, say  $W_j^*$  for  $j \in \{3, 4\}$ , then set  $n_j^{q_2} = |G_{q_2}^{**} \cap W_j^*| - 1$  and  $n_{7-j}^{q_2} = 1$ . Otherwise, without loss of generality, we choose  $v_3^{q_2}$  to be the end of  $G_{q_2}^{**}$  in  $W_3^*$ , and  $v_4^{q_2}$  to be the end of  $G_{q_2}^{**}$  in  $W_4^*$ . Then we set  $n_3^{q_2} = |G_{q_2}^{**} \cap W_3^*|$  and  $n_4^{q_2} = |G_{q_2}^{**} \cap W_4^*|$ .

Use Corollary 5.12 to distribute the vertices between  $W_2$ ,  $W_3$ ,  $W_4$ . We have finished distributing  $B_{q_2}$ .

Once all vertices from  $H_{i^*}$  have been assigned, we have the desired partition. The proof is complete.  $\square$

**Theorem 5.25.** *There exists an algorithm with the following specifications:*

**Input:** A 2-connected graph  $G$  on  $n$  vertices and  $m$  edges, four distinct vertices  $v_1, v_2, v_3, v_4$ , and natural numbers  $n_1, n_2, n_3, n_4$  such that  $G$  is 4-connected to  $\{v_1, v_2, v_3, v_4\}$  and  $n_1 + n_2 + n_3 + n_4 = n$ .

**Output:** A partition of  $V(G)$  into  $V_1, V_2, V_3, V_4$ , such that, for  $i = 1, 2, 3, 4$ ,  $G[V_i]$  is connected,  $v_i \in V_i$  and  $|V_i| = n_i$ .

**Running time:**  $O(n^3m)$

*Proof.* If  $|V(G) = 4|$ , then use the only possible partition,  $W_i = \{v_i\}$  for  $i = 1, 2, 3, 4$ . Otherwise, create a graph  $G'$  by adding a cycle between  $v_1, v_2, v_3, v_4$ , as well as a new vertex  $r$  whose neighbors are exactly  $v_1, v_2, v_3$ , and  $v_4$ . Apply Theorem 5.21 to construct a nonseparating chain decomposition  $(H_1, \dots, H_t)$  of  $G'$  rooted at  $r$  in  $O(n^2m)$  time. Permute indices so that  $v_1$  and  $v_2$  are in the first chain and  $v_3$  and  $v_4$  are in the last chain.

Let  $i^*$  be the largest index such that  $\left| G \left[ \bigcup_{j=1}^{i^*-1} I(H_j) \right] \right| \leq n_1 + n_2 + 1$ , and call the difference  $n^* = (n_1 + n_2 + 1) - \left| G \left[ \bigcup_{j=1}^{i^*-1} I(H_j) \right] \right|$ .

If  $n^* = 0$ , Theorem 5.9 to  $G \left[ \bigcup_{j=1}^{i^*-1} I(H_j) \right]$  to get  $W_1$  and  $W_2$ , then again to  $G \left[ \bigcup_{j=i^*}^t I(H_j) \right]$  to get  $W_3$  and  $W_4$ . This takes  $O(m)$  time.

If  $n^* > 0$ , break into cases on the chain type of  $H_{i^*}$  (it cannot be an elementary chain)

- Suppose  $H_{i^*}$  is a triangle chain with vertices  $x_1, x_2, x_3, w_1, w_2, w_3$  and neighbors  $y_1, y_2, y_3$ . If  $n^* = 2$ , complete the following steps in the reversed chain decomposition. Create a new ear  $G^*$ , having 1 internal vertex and ends chosen from  $x_1, x_2, x_3$  arbitrarily. Apply Theorem 5.9 to  $G \left[ V(G^*) \cup \bigcup_{j=1}^{i^*-1} I(H_j) \right]$  to get a tentative  $W_1^*$  and  $W_2^*$ . To get  $W_1$  and  $W_2$ , we replace the internal vertex of  $G^*$  with  $w_1$ . Apply Theorem 5.9 to  $G \left[ \bigcup_{j=i^*}^t I(H_j) \right]$  to get  $W_3$  and  $W_4$ . This takes  $O(m)$  time.

- Suppose  $H_{i^*} = b_0 B_1 b_1 \dots b_{k-1} B_\ell b_\ell$  is an up chain or down chain. If  $G_{i^*}$  is a down chain, complete the following steps in the reversed chain decomposition. Create a new ear  $G^*$ , having  $n^*$  internal vertices and ends  $b_0, b_\ell$ . Apply Theorem 5.9 to  $G \left[ V(G^*) \cup \bigcup_{j=1}^{i^*-1} I(H_j) \right]$  to get a tentative  $W_1^*$  and  $W_2^*$ .

We introduce some notation for  $H_{i^*}$ , organizing its pieces in the same way that we organized the chains above. Suppose we have assigned  $p_1$  internal vertices of  $H^*$  to  $W_1^*$ , and the remaining  $p_2$  to  $W_2^*$ . Without loss of generality,  $p_1 > 0$  so there is an end of  $H^*$  in  $W_1$ , say  $b_0$ .

Let  $q_1$  be the largest index so that  $|V(B_1 \cup B_2 \cup \dots \cup B_{q_1-1}) - \{b_0, b_\ell\}| \leq p_1$ , and call the difference  $n_1^* = p_1 - |V(B_1 \cup B_2 \cup \dots \cup B_{q_1-1}) - \{b_0, b_\ell\}|$ . Similarly, let  $q_2$  be the minimal index so that  $|V(B_{q_2+1} \cup B_{q_2+2} \cup \dots \cup B_\ell) - \{b_0, b_\ell\}| \leq p_2$ , and call the difference  $n_2^* = p_2 - |V(B_{q_2+1} \cup B_{q_2+2} \cup \dots \cup B_\ell) - \{b_0, b_\ell\}|$ .

Assign every vertex from  $W_j^* - G^*$  to  $W_j$  for  $j = 1, 2$ . Further assign  $\left( \bigcup_{j=1}^{q_1-1} V(B_j) \right) - \{b_\ell\}$  to  $W_1$  and  $\left( \bigcup_{j=q_2+1}^\ell V(B_j) \right) - \{b_0\}$  to  $W_2$ .

Build an open ear decomposition rooted at  $r$  on  $G - \left( \bigcup_{j=1}^{i^*} I(H_j) - \{r\} \right)$  in  $O(m)$  time, then extend it with new ears as follows. Let  $I$  be the set of all indices  $i \in \{q_1, q_1 + 1, \dots, q_2\}$  such that either  $B_i$  is 2-connected, or else  $i > q_1$  and  $B_i, B_{i-1}$  are each induced by an edge. For each  $i \in I$ , create an ear  $G_i^{**}$  with the following specifications:

- If  $B_i$  is two-connected, choose the ends of  $G_i^{**}$  to be the two vertices in  $\mathcal{N}(B_i) \cap (V(G') - V(F - r))$ . If not, then choose any two vertices in  $\mathcal{N}(b_i) \cap (V(G') - V(F - r))$ .
- Create

$$|B_i| - \mathbb{1}_{i+1 \in I} - (n_1^* + 1) \cdot \mathbb{1}_{i=q_1} - (n_2^* + 1) \cdot \mathbb{1}_{i=q_2}$$

internal vertices in  $G_i^{**}$ .

Apply Theorem 5.9 to  $G \left[ \left( \bigcup_{i \in I} V(G_i^{**}) \right) \cup \left( \bigcup_{j=i^*+1}^t I(H_j) \right) \right]$  to get a tentative  $W_3^*$  and  $W_4^*$ . This takes  $O(m)$  time.

Assign every vertex from  $W_j^* - \bigcup_{i \in I} G_i^{**}$  to  $W_j$  for  $j = 3, 4$ .

If  $q_1 = q_2$ , apply the algorithm recursively, using the boundary vertices of  $B_{q_1} = B_{q_2}$  as  $v_1, v_2, v_3, v_4$ , to distribute  $B_{q_1} = B_{q_2}$  among  $W_1, W_2, W_3, W_4$ . As each level of the recursion takes  $O(n^2m)$  time (limited by creating the nonseparating chain decomposition) and there are at most  $n$  levels, this step takes  $O(n^3m)$  time. We have the desired partition.

If  $q_1 < q_2$ , iterate through indices in  $I$ .

– If  $q_1 \in I$ ,

- \* Set  $v_1^{q_1} = b_{q_1-1}$  and  $n_1^{q_1} = n_1^* + 1$ .
- \* If  $q_1 + 1 \in I$ , set  $v_2^{q_1} = b_{q_1}^{q_1}$  and  $n_2^{q_1} = 1$ .
- \* Use the ends of  $G_{q_1}^{**}$  as  $v_3^{q_1}$  and  $v_4^{q_1}$ . If all of  $G_{q_1}^{**}$  was assigned to the same tentative part, say  $W_j^*$  for  $j \in \{3, 4\}$ , then set  $n_j^{q_1} = |G_{q_1}^{**} \cap W_j^*| - 1$  and  $n_{7-j}^{q_1} = 1$ . Otherwise, without loss of generality, we choose  $v_3^{q_1}$  to be the end of  $G_{q_1}^{**}$  in  $W_3^*$ , and  $v_4^{q_1}$  to be the end of  $G_{q_1}^{**}$  in  $W_4^*$ . Then we set  $n_3^{q_1} = |G_{q_1}^{**} \cap W_3^*|$  and  $n_4^{q_1} = |G_{q_1}^{**} \cap W_4^*|$ .

Distribute the vertices between  $W_1, W_3$ , and  $W_4$  using either Corollary 5.12 (if  $i + 1 \notin I$ ) in  $O(m)$  time, or recursively (if  $i + 1 \in I$ ). As each level of the recursion takes  $O(n^2m)$  time (limited by creating the nonseparating chain decomposition) and there are at most  $n$  levels, this step takes  $O(n^3m)$  time.

– For each  $i \in I - \{q_1, q_2\}$ , check whether all of  $G_i^{**}$  was assigned to the same tentative part, say  $W_j^*$  for  $j \in \{3, 4\}$ . If so, then assign every vertex in  $B_i - b_i$  to  $W_j$ . If  $i + 1 \notin I$ , further assign  $b_i$  to  $W_j$ . Otherwise,

- \* If  $i + 1 \in I$ , set  $v_2^i = b_i$  and  $n_2^i = 1$ .

- \* Set the end of  $G_i^{**}$  in  $W_3$  as  $v_3^i$ .
- \* Set the end of  $G_i^{**}$  in  $W_4$  as  $v_4^i$ .

Distribute the vertices between  $W_3$  and  $W_4$  using either Theorem 1.4 with  $k = 2$  (if  $i + 1 \notin I$ ) or Corollary 5.12 (if  $i + 1 \in I$ ), in  $O(m)$  time either way.

– For  $B_{q_2}$ , we distribute vertices as follows.

- \* Set  $v_2^{q_2} = b_{q_2}$  and  $n_2^{q_2} = n_2^* + 1$ .
- \* Use the ends of  $G_{q_1}^{**}$  as  $v_3^{q_2}$  and  $v_4^{q_2}$ . If all of  $G_{q_2}^{**}$  was assigned to the same tentative part, say  $W_j^*$  for  $j \in \{3, 4\}$ , then set  $n_j^{q_2} = |G_{q_2}^{**} \cap W_j^*| - 1$  and  $n_{7-j}^{q_2} = 1$ . Otherwise, without loss of generality, we choose  $v_3^{q_2}$  to be the end of  $G_{q_2}^{**}$  in  $W_3^*$ , and  $v_4^{q_2}$  to be the end of  $G_{q_2}^{**}$  in  $W_4^*$ . Then we set  $n_3^{q_2} = |G_{q_2}^{**} \cap W_3^*|$  and  $n_4^{q_2} = |G_{q_2}^{**} \cap W_4^*|$ .

Use Corollary 5.12 to distribute the vertices between  $W_2, W_3, W_4$  in  $O(m)$  time.

Once we have iterated through all of  $I$ , we have the desired partition.

□

## REFERENCES

- [1] K. Ando, H. Enomoto, and A. Saito, “Contractible Edges in 3-Connected Graphs,” *J. Comb. Theory B*, vol. 42, pp. 87–93, 1987.
- [2] L. S. Chandran, Y. K. Cheung, and D. Isaac, “Spanning Tree Congestion and Computation of Generalized Györi-Lovász Partition,” *arXiv e-prints*, 2018. arXiv: 1802.07632.
- [3] J. Chen, R. D. Kleinberg, L. Lovász, R. Rajaraman, R. Sundaram, and A. Vetta, “(Almost) Tight Bounds and Existence Theorems for Single-Commodity Confluent Flows,” *J. ACM*, vol. 54(4), 2007, Article 16.
- [4] J. Cheriyan and S. N. Maheshwari, “Finding Nonseparating Induced Cycles and Independent Spanning Trees in 3-Connected Graphs,” *J. Algorithms*, vol. 9, pp. 507–587, 1988.
- [5] S. Curran, O. Lee, and X. Yu, “Chain Decompositions of 4-Connected Graphs,” *SIAM J. Discrete Math*, vol. 19(4), pp. 848–880, 2005.
- [6] ———, “Nonseparating planar chains in 4-connected graphs,” *SIAM J. Discrete Math*, vol. 19(2), pp. 399–419, 2005.
- [7] ———, “Finding four independent trees,” *SIAM J. Comput.*, vol. 35(5), pp. 1023–1058, 2006.
- [8] S. Evan and R. E. Tarjan, “Computing an *st*-Numbering,” *Theor. Comput. Sci.*, vol. 2(3), pp. 339–344, 1976.
- [9] A. Frank, “Combinatorial algorithms, algorithmic proofs,” in Hungarian, PhD thesis, Eötvös University, Budapest, 1976.
- [10] A. Gopalan and S. Ramasubramanian, “On constructing three edge independent spanning trees,” Available at [citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.442.3598&rep=rep1&type=pdf](https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.442.3598&rep=rep1&type=pdf).
- [11] E. Györi, “On division of graphs to connected subgraphs,” *Colloq. Math. Soc. János Bolyai*, vol. 18, pp. 485–494, 1976, Volume Title: Combinatorics, Keszthely (Hungary).
- [12] A. Huck, “Independent trees in graphs,” *Graphs and Combinatorics*, vol. 10, pp. 29–45, 1994.

- [13] A. Itai and M. Rodeh, “The multi-tree approach to reliability in distributed networks,” in *Proceedings 25th Annual IEEE Sympos. on Fund. of Comput. Sci.*, 1984, pp. 137–147.
- [14] K. Kawarabayashi, O. Lee, and X. Yu, “Non-Separating Paths in 4-Connected Graphs,” *Ann. Comb.*, vol. 9, pp. 47–56, 2005.
- [15] S. Khuller and B. Schieber, “On Independent Spanning Trees,” *Information Processing Letters*, vol. 42, pp. 321–323, 1992.
- [16] L. Lovász, “A homology theory for spanning trees of a graph,” *Acta Math. Acad. Sci. Hungaricae*, vol. 30, pp. 241–251, 1977.
- [17] W. Mader, “A reduction method for edge-connectivity in graphs,” *Advances in Graph Theory*, vol. 3, pp. 145–164, 1978.
- [18] L. Schlipf, “The Mondschein sequence,” in *Proceedings 41st Intl. Colloq. on Automata, Languages, and Programming*, 2014, pp. 967–978.
- [19] L. Schlipf and J. M. Schmidt, “Edge-Orders,” in *Proceedings 44th Intl. Colloq. on Automata, Languages, and Programming*, 2017, 75:1–75:14.
- [20] A. Zehavi and A. Itai, “Three tree-paths,” *J. Graph Theory*, vol. 13, pp. 175–188, 1989.

## VITA

Alexander Hoyer was born and raised in Menomonee Falls, Wisconsin. In 2009, he graduated from Rufus King International High School in Milwaukee, Wisconsin, earning an International Baccalaureate diploma. In 2013, he graduated from Drake University in Des Moines, Iowa, earning a B.S. in Mathematics and a B.S.B.A. in Actuarial Science. In 2019, he graduated from the Georgia Institute of Technology in Atlanta, Georgia, earning a Ph.D. in Mathematics, after studying structural graph theory as an advisee of Dr. Robin Thomas.

As a researcher, he produced one publication and made three professional presentations. As a teaching assistant and graduate student instructor, he earned department accolades twice for outstanding student evaluations, once as an outstanding teaching assistant, and once as an outstanding graduate student instructor.